

IMPLEMENTASI SISTEM PENDETEKSI *PREMATURE VENTRICULAR CONTRACTION (PVC)* ARITMIA MENGUNAKAN METODE *NAÏVE BAYES*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Gusti Arief Gilang
NIM: 145150301111053



PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

IMPLEMENTASI SISTEM PENDETEKSI *PREMATURE VENTRICULAR CONTRACTION (PVC)* ARITMIA MENGGUNAKAN METODE *NAÏVE BAYES*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Gusti Arief Gilang
NIM: 145150301111053

Skripsi ini telah diuji dan dinyatakan lulus pada
29 Juni 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Rizal Maulana, S.T., M.T., M.Sc.

NIK. 201607 891009 1 001

Dosen Pembimbing II



Wijaya Kurniawan, S.T., M.T

NIP. 19820125 201504 1 002

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D

NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang tertulis dalam dan disebutkan dalam daftar pustaka pada naskah ini.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 Ayat 2 dan Pasal 70).

METERAI
TEMPEL

250CCAFF170025462

6000
ENAM RIBU RUPIAH

Malang, 20 Mei 2018


Gusti Arief Gilang

NIM. 145150301111053

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah Subhanallahu wa Ta'ala karena berkah rahmat dan hidayah-Nya, penulis dapat menyelesaikan penelitian dan Laporan Skripsi untuk memperoleh gelar Sarjana Teknik yang berjudul **Implementasi Sistem Pendeteksi *Premature Ventricular Contraction* (PVC) Aritmia Berdasarkan Waktu Interval Detak Jantung dan BPM Menggunakan Metode *Naïve Bayes* Berbasis Arduino Uno.**

Dalam pelaksanaan dan penyusunan laporan skripsi ini, tidaklah sedikit hambatan dan kesulitan yang penulis hadapi. Namun, penulis menyadari bahwa kelancaran dalam penyusunan laporan ini tidak lain adalah berkat doa, bantuan dan bimbingan dari berbagai pihak, sehingga kendala-kendala yang penulis hadapi dapat teratasi. Penghargaan dan terima kasih yang sebesar-besarnya penulis sampaikan kepada:

1. Bapak Alit Jaya Sumantri, Ibu Sri Rahayu, Eka Shuva Prabu Wicaksana dan Nada Syifa Fauziah selaku kedua orang tua dan saudara-saudari kandung dari penulis yang tidak henti memberikan doa, semangat dan materi kepada penulis.
2. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D selaku Ketua Jurusan Teknik Informatika periode 2016 – saat ini.
3. Bapak Sabriansyah Rizqika Akbar, S.T., M. Eng. selaku Ketua Program Studi Teknik Komputer periode 2016 – saat ini.
4. Bapak Rizal Maulana, S.T., M.T., M.Sc. selaku dosen Pembimbing I yang selalu membantu memberikan bimbingan dan arahan selama pengerjaan skripsi.
5. Bapak Wijaya Kurniawan, S.T, M.T. selaku dosen Pembimbing II yang selalu membantu memberikan bimbingan dan arahan selama pengerjaan skripsi.
6. Muhammad Taufiq Firmansyah, Muhamad Naufal, dan Nafisa yang selalu mengisi hari-hari penulis dengan canda tawa sehingga penulis tetap semangat dalam menyelesaikan penulisan skripsi ini.
7. Teman-teman PSDM HIMATEKKOM, NI Squad, Kontrakan Pojok, *Most Wanted Rampal Basketball*, *XILINX circle*, serta teman-teman Teknik Komputer 2014 yang selalu memberikan motivasi kepada penulis untuk menyelesaikan penulisan skripsi ini.
8. Teman-teman grup Anak Shaleh yang selalu memberikan konten yang bermanfaat sehingga penulis tetap semangat dalam melakukan penelitian.
9. Seluruh pihak yang tidak dapat disebutkan satu persatu yang telah berperan dalam penyelesaian skripsi ini.

Akhir kata penulis menyadari bahwa dalam penulisan skripsi ini masih terdapat banyak kekurangan yang perlu diperbaiki. Karena itu, kritik dan saran yang

membangun sangat diharapkan oleh penulis. Semoga kedepannya skripsi ini dapat bermanfaat bagi semua pihak yang membacanya.

Malang, 20 Mei 2018

Penulis

gusti.arief@outlook.com



ABSTRAK

Premature Ventricular Contraction (PVC) Aritmia merupakan kelainan detak jantung akibat dari gangguan ritme jantung di bagian ventrikel. PVC yang terjadi terlalu sering kepada seseorang dapat berlanjut ke penyakit yang membahayakan seperti gagal jantung. Namun saat ini, untuk melakukan deteksi kelainan detak jantung PVC ini hanya dapat dilakukan di rumah sakit. Dari permasalahan tersebut, diperlukan penelitian untuk mendeteksi jenis PVC aritmia yang dapat digunakan secara mandiri dan tidak memakan biaya yang besar sehingga dapat mengurangi individu yang memiliki penyakit jantung yang terjadi akibat PVC. Penelitian ini memiliki parameter-parameter yang digunakan untuk membandingkan jenis PVC yaitu waktu interval detak jantung manusia dan BPM dengan menggunakan teknik EKG untuk membaca detak jantung tersebut. Penentuan perbandingan jenis PVC dengan parameter R Interval dan BPM diperoleh dari hasil perhitungan pembacaan sensor EKG AD8232 oleh Arduino Uno menggunakan metode *Naïve Bayes*. Metode *Naïve Bayes* digunakan dalam penelitian ini karena memiliki kelas jenis PVC aritmia yang diketahui sejak awal. Pengujian untuk menentukan nilai BPM pada sistem menghasilkan persentase rata-rata *error* sebesar 8,558%. Selain itu, pengujian hasil klasifikasi sistem menggunakan metode *Naïve Bayes* dengan 46 data latih dan 14 data uji menghasilkan akurasi 92,857% dengan waktu komputasi rata-rata 7,2 detik.

Kata kunci: EKG, aritmia, *PVC*, klasifikasi, *Naïve Bayes*

ABSTRACT

Premature Ventricular Contraction (PVC) Arrhythmia is an anomaly heartbeat that occurs because of the heartbeat rhythm disorder in the ventricles. PVC that happens too often can lead to dangerous disease such as heart failure. At this time, to detect the PVC can only be done at the hospital. From this problem, it is necessary to make a system that detect the type of PVC that can be used independently so it can reduce the number of individuals who have a heart disease because of PVC. This research has a parameter that used to make a comparison on each type of PVC, that is heartbeat time interval and BPM by using ECG techniques to measure the heartbeat. Determination of PVC type with heartbeat interval and BPM parameter is obtained from AD8232 sensor value by Arduino Uno using Naïve Bayes method. Naïve Bayes is used in this research because it has a class type of PVC arrhythmia that known from the beginning. A BPM tests of this research gives 8,558% percentage average error. Furthermore, a classification result test using the Naïve Bayes method with 46 training data and 14 test data gives an accuration 92,857% with the average computation time 7,2 second.

Keywords: ECG, arrhythmia, PVC, classification, Naïve Bayes



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian	2
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori	7
2.2.1 Elektrokardiografi.....	7
2.2.2 Type equation here.Aritmia.....	9
2.2.3 Mikrokontroler Arduino Uno	10
2.2.4 EKG Module AD8232 Arduino.....	12
2.2.5 <i>Naïve Bayes Algorithm</i>	13
2.2.6 Pemrograman C Arduino.....	15
2.2.7 Pemrograman Java Processing.....	16
BAB 3 METODE PENELITIAN	18
3.1 Alur Metodologi Penelitian.....	18
3.2 Studi Pengkajian Literatur.....	19
3.3 Analisis Kebutuhan Sistem.....	19
3.3.1 Kebutuhan Perangkat Lunak	20
3.3.2 Kebutuhan Perangkat Keras.....	20
3.4 Pengumpulan Data.....	20

3.5	Perancangan Sistem.....	20
3.5.1	Perancangan Perangkat Lunak.....	20
3.5.2	Perancangan Perangkat Keras	21
3.6	Implementasi Sistem.....	22
3.7	Pengujian dan Analisis	22
3.8	Kesimpulan dan Saran.....	22
BAB 4 REKAYASA KEBUTUHAN		24
4.1	Gambaran Umum Sistem.....	24
4.2	Analisis Kebutuhan Sistem	24
4.2.1	Kebutuhan Fungsional	24
4.2.2	Kebutuhan Non-Fungsional	25
4.3	Batasan Desain Sistem	26
BAB 5 PERANCANGAN DAN IMPLEMENTASI		27
5.1	Perancangan Sistem.....	27
5.1.1	Perancangan <i>Prototype</i> Sistem	27
5.1.2	Perancangan Perangkat Keras Sistem	27
5.1.3	Perancangan Perangkat Lunak Sistem	30
5.2	Implementasi Sistem.....	45
5.2.1	Implementasi <i>Prototype</i> Sistem	45
5.2.2	Implementasi Perangkat Keras Sistem.....	46
5.2.3	Implementasi Perangkat Lunak Sistem	47
BAB 6 PENGUJIAN DAN ANALISIS.....		70
6.1	Pengujian Akurasi Perhitungan BPM	70
6.1.1	Tujuan Pengujian.....	70
6.1.2	Prosedur Pengujian	70
6.1.3	Hasil dan Analisis Pengujian.....	72
6.2	Pengujian Akurasi Hasil Klasifikasi <i>Naïve Bayes</i>	73
6.2.1	Tujuan Pengujian.....	73
6.2.2	Prosedur Pengujian	73
6.2.3	Hasil dan Analisis Pengujian.....	74
6.3	Pengujian Waktu Komputasi Pemrosesan Sistem	76
6.3.1	Tujuan Pengujian.....	77
6.3.2	Prosedur Pengujian	77
6.3.3	Hasil dan Analisis Pengujian.....	78
BAB 7 PENUTUP		80
7.1	Kesimpulan.....	80

7.2	Saran	80
DAFTAR PUSTAKA.....		81



DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka	6
Tabel 2.2 Jenis <i>PVC</i> Aritmia	10
Tabel 2.3 Spesifikasi Arduino Uno.....	11
Tabel 2.4 Deskripsi Pin Sensor EKG Module AD8232.....	12
Tabel 5.1 Keterangan Pin Sensor AD8232.....	28
Tabel 5.2 Keterangan Koneksi Pin Sensor AD8232 Dengan Arduino Uno	29
Tabel 5.3 Penempatan Elektroda	29
Tabel 5.4 Nilai <i>Mean</i> Pada Setiap Jenis Fitur dan Kelas Aritmia	39
Tabel 5.5 Nilai Standar Deviasi Pada Setiap Jenis Fitur dan Kelas Aritmia	41
Tabel 5.6 Kode Program Inisialisasi <i>library</i> Sistem Pendeteksi <i>Premature Ventricular Contraction</i> Aritmia	47
Tabel 5.7 Kode Program Inisialisasi Variabel Pembacaan Sensor.....	48
Tabel 5.8 Kode Program Pembacaan Nilai Sensor AD8232	49
Tabel 5.9 Kode Program Mencari Nilai Titik Terendah dan Titik Tertinggi Gelombang.....	50
Tabel 5.10 Kode Program Mencari Nilai R Interval dan BPM	51
Tabel 5.11 Kode Program Penentuan Nilai R Interval dan BPM Sebagai Fitur Klasifikasi <i>Naïve Bayes</i>	52
Tabel 5.12 Kode Program Inisialisasi Variabel <i>Naïve Bayes</i>	54
Tabel 5.13 Kode Program Inisialisasi Variabel <i>Prior, Mean</i> dan Standar Deviasi .	54
Tabel 5.14 Kode Program Fungsi <i>Gaussian</i>	55
Tabel 5.15 Kode Program Fungsi <i>ProbabilityPosterior</i>	56
Tabel 5.16 Kode Program Penentuan Kesimpulan	56
Tabel 5.17 Kode Program Pengiriman Data dari Arduino ke Processing.....	57
Tabel 5.18 Kode Program Inisialisasi Variabel Pada Processing	59
Tabel 5.19 Kode Program Pengaturan Tampilan Awal Pada Processing	60
Tabel 5.20 Kode Program Pengambilan Data Dari Arduino Ke Processing.....	62
Tabel 5.21 Kode Program Menampilkan Grafik EKG Pada Processing	64
Tabel 5.22 Kode Program Menampilkan Nilai Fitur Pada Processing.....	65
Tabel 5.23 Kode Program Menampilkan Hasil Kesimpulan Klasifikasi Metode <i>Naïve Bayes</i> Pada Processing	68

Tabel 6.1 Hasil Pengujian Perbandingan Nilai Pendeteksian BPM Manual dengan Pendeteksian BPM oleh Sistem	72
Tabel 6.2 Data dan Hasil Pengujian Klasifikasi <i>Naïve Bayes</i>	74
Tabel 6.3 Nilai <i>Gaussian</i> Pada Setiap Jenis Fitur dan Kelas Aritmia.....	75
Tabel 6.4 Implementasi Kode Program Arduino Untuk Menghitung Waktu Komputasi Sistem.....	77
Tabel 6.5 Hasil Pengujian Waktu Komputasi Sistem.....	78



DAFTAR GAMBAR

Gambar 2.1 Aktivitas Listrik Jantung.....	7
Gambar 2.2 Penempatan Elektroda Menggunakan Teknik Lead	8
Gambar 2.3 Pola EKG Normal	9
Gambar 2.4 Arduino Uno	11
Gambar 2.5 Sensor EKG AD8232.....	12
Gambar 2.6 Arduino IDE	16
Gambar 2.7 Perangkat Lunak Processing.....	17
Gambar 3.1 Diagram Alir Metode Penelitian.....	18
Gambar 3.2 Diagram Alir Perancangan Perangkat Lunak.....	21
Gambar 3.3 Blok Diagram Sistem	22
Gambar 5.1 Desain <i>Prototype</i> Sistem	27
Gambar 5.2 Skematik Perancangan Perangkat Keras.....	28
Gambar 5.3 Penempatan Elektroda pada Sistem Pendeteksi <i>Premature Ventricular Contraction</i> Aritmia.....	29
Gambar 5.4 Diagram Alir Perancangan Perangkat Lunak Pengambilan Data Sensor	30
Gambar 5.5 Diagram Alir Perancangan Penentuan Nilai R Interval dan BPM	32
Gambar 5.6 Diagram Alir Perancangan Klasifikasi <i>Naïve Bayes</i>	34
Gambar 5.7 Diagram Alir fungsi ProbabilityPrior()	35
Gambar 5.8 Diagram Alir Fungsi Gaussian()	35
Gambar 5.9 Diagram Alir Fungsi ProbabilityPosterior().....	36
Gambar 5.10 Diagram Alir Pengiriman Data Arduino ke Processing.....	42
Gambar 5.11 Diagram Alir Menampilkan Hasil Keluaran Menggunakan Processing	44
Gambar 5.12 Implementasi <i>Prototype</i> Alat Pendeteksi <i>Premature Ventricular Contraction</i> Aritmia.....	45
Gambar 5.13 Implementasi Perangkat Keras Sistem.....	46
Gambar 5.14 Implementasi Penempatan Elektroda	47
Gambar 5.15 Tampilan Awal Sistem Menggunakan Processing.....	62
Gambar 5.16 Tampilan Grafik EKG Pada Processing.....	65
Gambar 5.17 Tampilan Fitur R Interval Pada Processing.....	67
Gambar 5.18 Fitur R Interval pada Grafik EKG.....	67

Gambar 5.19 Tampilan Fitur BPM Pada Processing.....	68
Gambar 5.20 Tampilan Hasil Kesimpulan Klasifikasi <i>Naïve Bayes</i> Pada Processing	69
Gambar 6.1 Pendeteksian BPM Menggunakan Deteksi Manual	71
Gambar 6.2 Pendeteksian BPM Menggunakan Deteksi dari Sistem	71
Gambar 6.3 Hasil Pengujian Waktu Komputasi Sistem	79



DAFTAR LAMPIRAN

LAMPIRAN A DATA SET	82
LAMPIRAN B DATA UJI	84
LAMPIRAN C KODE PROGRAM KESELURUHAN SISTEM	85
C.1 Kode Program Arduino	85
C.1.1 Kode Program Utama.....	85
C.1.2 Kode Program Klasifikasi <i>Naïve Bayes</i>	89
C.2 Kode Program Processing	92



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Penyakit jantung merupakan salah satu penyakit yang berbahaya bagi setiap orang. Penyakit jantung ini merupakan penyebab nomor satu kematian terbanyak selama 15 tahun terakhir. Dari 56 juta kematian di dunia pada tahun 2015, sekitar 14% kematian terjadi karena penyakit jantung (World Health Organization, 2017). Kematian akibat dari penyakit jantung biasanya terjadi karena beban jantung yang terlalu berat sehingga mengalami serangan jantung mendadak.

Salah satu jenis kelainan yang terjadi pada jantung, yaitu Aritmia merupakan gejala yang paling banyak mengakibatkan kematian pada pasien yang memiliki penyakit ginjal kronis. Penyakit ginjal kronis tersebut memberikan beban yang berat kepada jantung sehingga jantung mengalami Aritmia, lebih tepatnya ke dalam jenis *Premature Ventricular Contraction (PVC)*. Terhitung dari jumlah kematian pasien penyakit ginjal kronis pada tahun 2008, sekitar 30% diantaranya terjadi karena PVC (Charles A. Herzag, 2008). Aritmia Ventrikular merupakan kelainan detak jantung yang terjadi akibat dari gangguan perubahan kecepatan denyut pada jantung di bagian ventrikel (Departemen Kardiologi & Kedokteran Vaskular FKUI, 2004). Kemudian PVC merupakan jenis dari Aritmia Ventrikular, dimana terdapat tambahan detak jantung yang mengakibatkan irama detak jantung terganggu (Mayoclinic, 2018).

Di Indonesia, jumlah kematian akibat penyakit jantung mengalami peningkatan. Menurut data Fakultas Kedokteran Universitas Islam Indonesia pada tahun 2016, populasi penyakit jantung di Indonesia mencapai rata-rata 9,2%, dengan populasi tertinggi sebanyak 16,9% di Sulawesi Tengah dan populasi terendah di Lampung dengan angka 3,5%. Hal ini menunjukkan salah satu penyebab kematian akibat penyakit jantung adalah karena tidak meratanya dokter spesialis jantung ke seluruh daerah di Indonesia. Pernyataan ini didukung oleh Kepala Departemen Fisika Kedokteran FKUI Edrial Eddin yang menyebutkan bahwa di Indonesia dokter spesialis jantung tidak terdistribusi secara merata ke seluruh daerah sehingga banyak terjadi kasus penyakit jantung yang berujung pada kematian.

Pada saat ini, untuk mendeteksi aritmia yang akurat hanya dapat dilakukan oleh pihak rumah sakit. Baik itu menggunakan pemeriksaan EKG, pemeriksaan holter, pemeriksaan EPS (*Electrophysiology Study*) ataupun dengan melakukan MRI jantung (Fajar, 2017). Menurut DR. dr. Antonia Anna Lukito, SP.JP(K), FIHA, FAPSIC dari Siloam Hospitals Lippo Village, teknik yang paling akurat untuk mendeteksi penyakit jantung aritmia adalah menggunakan pemeriksaan MRI jantung karena dapat mengetahui apakah yang terdeteksi otot jantung atau lemak. Jika jantung terhalang/terhimpit oleh lemak maka kemungkinan terjadi aritmia sangat besar. Namun biaya untuk pemeriksaan MRI jantung ini memerlukan biaya sekitar 2-3 juta rupiah.

Terdapat banyak cara yang dapat digunakan untuk mengetahui kondisi jantung pada seseorang, salah satunya adalah dengan menggunakan teknik Elektrokardiografi (EKG). Saat ini teknik EKG banyak digunakan untuk mendeteksi suatu penyakit jantung karena teknik EKG dapat mengetahui kondisi jantung dengan menganalisa gelombang yang dihasilkan oleh aktivitas listrik jantung (Eduardo José da S.Luz, 2016). Selain itu, teknik EKG memungkinkan untuk menganalisa aktivitas listrik jantung secara real time. Dari alasan tersebut, penulis menggunakan teknik EKG untuk mengetahui kondisi jantung pada seseorang.

Dalam menentukan jenis dari PVC aritmia, diperlukan metode yang tepat untuk pengklasifikasiannya. Penulis menggunakan metode *Naïve Bayes* dalam menentukan jenis PVC Aritmia karena metode ini dapat melakukan klasifikasi dengan menggunakan fitur atau acuan yang dibutuhkan untuk menentukan jenis PVC Aritmia, yaitu waktu interval dari detak jantung dan nilai BPM. BPM (*Beat per Minute*) adalah jumlah detak jantung seseorang selama 1 menit. Selain itu, *Naïve Bayes* memiliki data latih dari jenis PVC aritmia yang sudah ada sebelumnya. Data latih tersebut dijadikan acuan untuk melakukan klasifikasi, dimana jenis dari PVC aritmia memiliki pola fitur yang tidak jauh berbeda sehingga dapat diklasifikasikan dengan tingkat kesalahan yang sedikit. Dengan alasan tersebut, penulis melakukan penelitian "Implementasi Sistem Pendeteksi Premature Ventricular Contraction (PVC) Aritmia Menggunakan Metode *Naïve Bayes*" sehingga pengguna dari sistem ini dapat mendeteksi penyakit jantung beserta gejalanya yang dapat digunakan secara mandiri dan tidak memakan biaya yang besar. Dengan begitu walaupun tanpa dokter spesialis, masyarakat dapat mengetahui kondisi jantung secara mandiri dan murah.

1.2 Rumusan Masalah

Dengan berlandaskan latar belakang yang dipaparkan di atas, maka dapat disimpulkan rumusan masalah sebagai berikut :

1. Bagaimana tingkat keakuratan pembacaan sensor EKG Module AD8232 Arduino terhadap detak jantung manusia?
2. Bagaimana tingkat keakuratan dari metode *Naïve Bayes* untuk mendiagnosa kelainan detak jantung *Premature Ventricular Contraction* Aritmia?
3. Berapa waktu yang dibutuhkan oleh sistem untuk menentukan hasil klasifikasi *Naïve Bayes* dari jenis kelas *Premature Ventricular Contraction* Aritmia?

1.3 Tujuan Penelitian

Tujuan yang ingin didapatkan dalam penulisan skripsi ini antara lain sebagai berikut :

1. Melakukan pengujian dan perhitungan tingkat keakuratan pembacaan sensor ECG Module AD8232 Arduino terhadap detak jantung manusia dengan melakukan perhitungan BPM.

2. Melakukan pengujian dan perhitungan tingkat keakuratan metode *Naïve Bayes* untuk diagnosa kelainan detak jantung aritmia.
3. Mengetahui waktu yang diperlukan oleh sistem untuk mendapatkan hasil klasifikasi.

1.4 Manfaat Penelitian

Manfaat yang dapat diperoleh dari penelitian ini antara lain sebagai berikut :

1. Untuk mengetahui kondisi jantung secara mudah dan mandiri kepada setiap individu.
2. Memberikan pengetahuan dan wawasan kepada pembaca mengenai penggunaan Module EKG AD8232 heartbeat heart rate arduino dan metode *Naïve Bayes* yang diterapkan dalam penelitian ini.

1.5 Batasan Masalah

Batasan masalah dibuat agar permasalahan yang dirumuskan tidak semakin melebar. Pada penelitian ini batasan masalah yang dibuat adalah :

1. Objek yang diamati adalah kondisi jantung pada manusia.
2. Faktor *Premature Ventricular Contraction* Aritmia yang diperhatikan adalah delapan nilai interval ketika jantung berdetak dan nilai bpm.

1.6 Sistematika Pembahasan

Uraian metodologi penelitian dari masing-masing bab pada penelitian ini dengan sistematika sebagai berikut :

BAB I Pendahuluan

Bab ini berisi mengenai latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dan sistematika penulisan dari "Implementasi Sistem Pendeteksi Penyakit Jantung *Premature Ventricular Contraction* Aritmia Menggunakan *Naïve Bayes*".

BAB II Tinjauan Pustaka dan Dasar Teori

Bab ini memiliki bahasan mengenai tinjauan pustaka dari penelitian-penelitian sebelumnya dan dasar teori yang mendukung dalam pembuatan sistem ini. Teori diambil dari jurnal, buku dan web yang dibuat oleh lembaga resmi.

BAB III Metode Penelitian

Bab ini membahas mengenai langkah-langkah yang dilakukan dalam melaksanakan penelitian ini, yaitu studi literatur, analisis kebutuhan sistem, perancangan dan implementasi sistem, pengujian dan analisis dan kesimpulan.

BAB IV Rekayasa Kebutuhan

Bab ini berisi mengenai penjelasan deskripsi umum dari sistem, rekayasa kebutuhan tampilan antarmuka sistem, kebutuhan perangkat keras dan perangkat

lunak, kebutuhan fungsional, kebutuhan komunikasi, kebutuhan *performance* sistem dan batasan *design* sistem.

BAB V Perancangan dan Implementasi Sistem

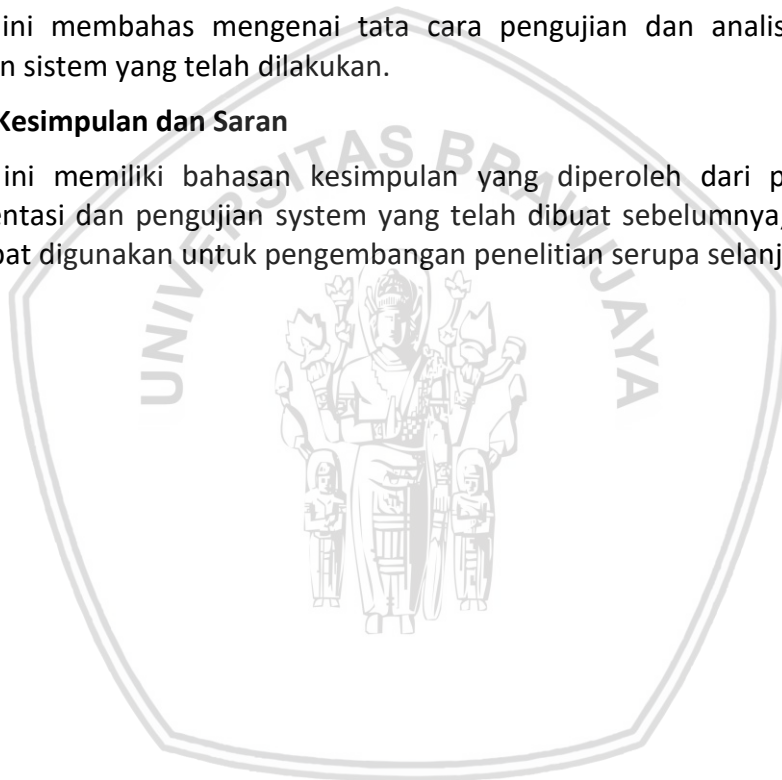
Bab ini membahas mengenai perancangan sistem untuk mendeteksi penyakit jantung *Premature Ventricular Contraction* Aritmia dengan metode *Naïve Bayes* dan proses implementasi sistem dalam mendeteksi penyakit jantung *Premature Ventricular Contraction* Aritmia tersebut. Proses dimulai dari membaca nilai sensor dengan mikrokontroler, proses nilai yang didapat menggunakan *Naïve Bayes*, kemudian menentukan kondisi jantung normal, *Premature Ventricular Contraction* Aritmia *bigeminy* atau *trigeminy*.

BAB VI Pengujian dan Analisis

Bab ini membahas mengenai tata cara pengujian dan analisa dari hasil pengujian sistem yang telah dilakukan.

BAB VII Kesimpulan dan Saran

Bab ini memiliki bahasan kesimpulan yang diperoleh dari perancangan, implementasi dan pengujian system yang telah dibuat sebelumnya, serta saran yang dapat digunakan untuk pengembangan penelitian serupa selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi tinjauan pustaka yang terdiri dari kajian pustaka dan dasar teori yang diperlukan dalam melakukan penelitian. Kajian pustaka membahas penelitian yang sudah ada sebelumnya dan memiliki kaitan dengan penelitian yang diusulkan, sedangkan dasar teori membahas mengenai berbagai teori yang diperlukan dalam menyusun penelitian ini.

2.1 Tinjauan Pustaka

Tinjauan pustaka membahas mengenai perbandingan antara penelitian sebelumnya dengan penelitian yang diusulkan oleh penulis. Teori-teori atau hasil dari penelitian-penelitian sebelumnya dijadikan acuan dan teori pendukung pada penelitian ini. Penelitian terdahulu yang difokuskan sebagai data acuan adalah yang berkaitan dengan metode untuk mendeteksi penyakit jantung aritmia. Oleh karena itu peneliti mengkaji hasil penelitian sebelumnya yang berupa jurnal-jurnal dari internet ataupun studi literatur.

Penelitian pertama dilakukan oleh Wahyu Kusuma R, Swelandiah E.P., Ridha I., dan Yasman R. dengan judul “Rancang Bangun Alat Telemonitor Aritmia Jantung Berbasis Web”. Penelitian tersebut memiliki tujuan untuk mengetahui hasil pengukuran rata-rata presisi waktu setiap beat dengan menggunakan *pulse sensor* dan menggunakan komunikasi internet yang kemudian diakses melalui halaman web. Selain itu, penelitian ini hanya memiliki keluaran berupa tidak terdeteksi aritmia dan terdeteksi aritmia. Hasil pengujian dari lima responden pada sistem ini memberikan nilai presisi rata-rata sebesar 86,52% (Wahyu Kusuma R, 2017).

Penelitian kedua dilakukan oleh K. Nanthakumar dan V. Suriyagayathri yang berjudul “*Prediction of Ventricular Arrhythmia using Naïve Bayes Classifier*”. Penelitian ini merupakan simulasi untuk mendeteksi *Premature Ventricular Contraction* Aritmia dengan klasifikasi *Naïve Bayes* menggunakan aplikasi *matlab*, dimana masukan yang digunakan adalah file .mat EKG yang berada di dalam MIT-BIH Database. Tujuan dari penelitian ini adalah untuk mencapai nilai akurasi yang tinggi dengan daya konsumsi yang rendah, yang didapatkan dengan membandingkan nilai frekuensi yang digunakan. Hasil dari penelitian ini adalah semakin kecil frekuensi yang digunakan, maka semakin kecil pula konsumsi daya yang digunakan. Frekuensi yang paling optimal adalah sebesar 10000 Hz dengan daya konsumsi sebesar 2,78 uW (K. Nanthakumar, 2017).

Dari alat yang telah dibuat pada penelitian pertama yang disebutkan di atas, terdapat kekurangan yaitu hanya mampu memberikan kesimpulan berupa terdeteksi aritmia atau tidak, dimana penyakit aritmia masih dapat diklasifikasikan menjadi berbagai macam jenis. Selain itu, pembacaan dari *pulse sensor* sangatlah minim karena tidak dapat mengetahui pola P-QRS-T detak jantung pada umumnya. Sementara itu, simulasi dari penelitian kedua hanya mampu membaca masukan berupa file dari MIT-BIH Database dan hanya memiliki satu jenis klasifikasi *Premature Ventricular Contraction* Aritmia yaitu *tachycardia arrhythmia*.

Dari kekurangan yang telah disebutkan di atas, penulis membuat Sistem Pendeteksi *Premature Ventricular Contraction* Aritmia Berdasarkan Waktu Interval Detak Jantung dan BPM Menggunakan Metode *Naïve Bayes* Berbasis Arduino ini. Dengan mempertimbangkan nilai masukan dari sensor berupa waktu interval dari jantung berdetak atau R interval, jumlah bpm yang didapatkan dan pola grafik EKG sehingga menambah nilai keakuratan sistem. Selain itu, sistem ini mendeteksi penyakit jantung *Premature Ventricular Contraction* Aritmia yang spesifik yaitu *trigeminy* dan *bigeminy*.

Adapun perbandingan penelitian yang diusulkan dengan penelitian yang terkait sebelumnya terdapat pada Tabel 2.1.

Tabel 2.1 Tinjauan Pustaka

No	Nama Penulis [Tahun], Judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1	Wahyu Kusuma R, Swelandiah E.P., Ridha L, Yasman R. [2017], Rancang Bangun Alat Telemonitor Aritmia Jantung Berbasis Web	Pembuatan alat untuk mendeteksi penyakit jantung aritmia dengan fitur berupa R interval	Perangkat untuk mendeteksi sinyal keluaran jantung adalah pulse sensor	Perangkat untuk mendeteksi sinyal keluaran jantung adalah sensor AD8232
			Metode yang digunakan adalah dengan melakukan perhitungan antara jarak R interval dengan algoritma aritmia (Webster, 1981).	Metode yang digunakan adalah dengan menggunakan metode <i>Naïve Bayes</i> , yaitu mengolah data uji sesuai pembacaan nilai sensor berdasarkan basis data latih.
2	K. Nanthakumar, V.	Klasifikasi penyakit jantung	Sistem yang digunakan	Mengimplementasikan alat dengan

	Suriyagayathri [2017], <i>Prediction of Ventricular Arrhythmia using Naïve Bayes Classifier</i>	aritmia dengan menggunakan metode <i>Naïve Bayes</i> .	merupakan simulasi dengan menggunakan matlab.	menggunakan Arduino dan sensor AD8232.
			Masukan dari sistem berupa data file matlab ekg.	Masukan dari alat berupa jantung manusia secara langsung.

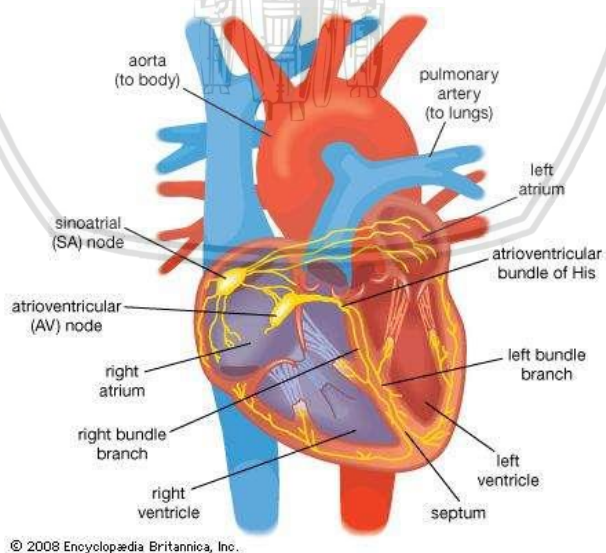
Sumber: (Wahyu Kusuma R, 2017), (K. Nanthakumar, 2017)

2.2 Dasar Teori

Dasar teori membahas mengenai berbagai macam teori yang diperlukan untuk menyusun penelitian yang diusulkan oleh penulis.

2.2.1 Elektrokardiografi

Elektrokardiografi (EKG) merupakan metode penggambaran grafik dari arus listrik yang dihasilkan oleh otot jantung setiap jantung berdetak (Encyclopedia Britannica, 2008). Hasil detak jantung dicatat dengan menggunakan elektrokardiogram dan memberikan informasi mengenai kondisi dan kinerja jantung. Aktivitas listrik dari jantung yang sehat di setiap individu dapat dijelaskan pada Gambar 2.1.

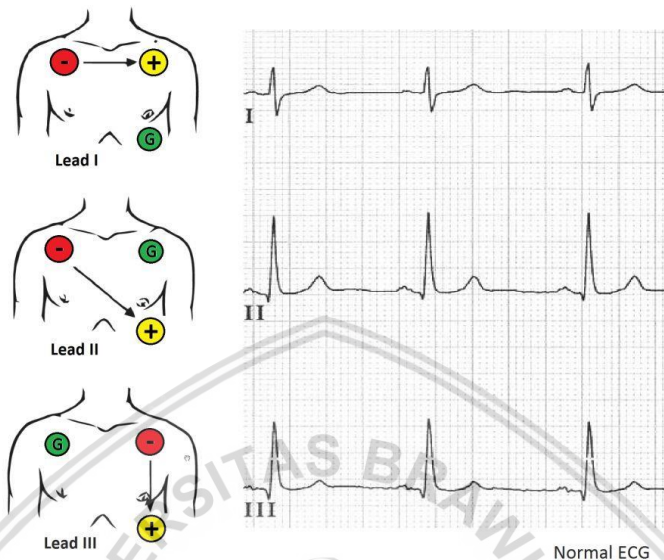


Gambar 2.1 Aktivitas Listrik Jantung

Sumber : (Encyclopedia Britannica, 2008)

Elektrokardiogram dibuat dengan menerapkan elektroda ke berbagai bagian tubuh. Salah satu teknik memasang elektroda yang merekam aktivitas listrik

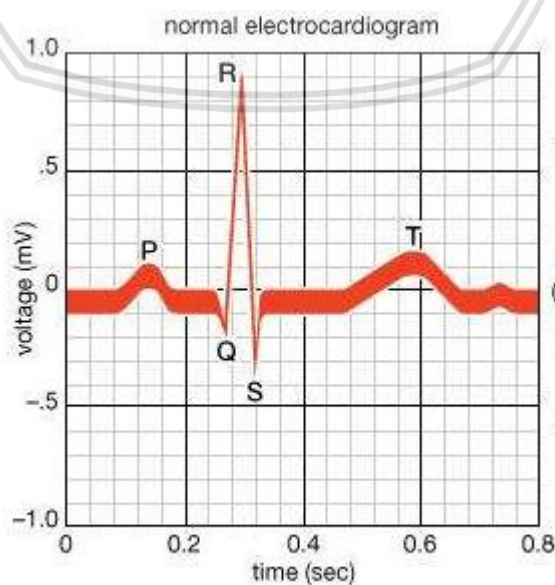
jantung pada bagian tubuh adalah dengan menggunakan teknik lead yang hanya memasang tiga elektroda ke tubuh. Teknik lead sendiri terdiri menjadi tiga macam, yaitu lead satu, lead dua dan lead tiga. Penempatan dari macam-macam lead tersebut dapat dilihat pada Gambar 2.2.



Gambar 2.2 Penempatan Elektroda Menggunakan Teknik Lead

Sumber : (Heart Risk Warner, 2015)

Setelah elektroda ditempatkan sesuai dengan teknik yang diinginkan, elektrokardiogram mengambil aliran listrik dari aktivitas jantung melalui perantara elektroda sehingga dapat terkalibrasi. Untuk mendapatkan hasil *standard* dari aktivitas listrik jantung biasanya digunakan dengan menggunakan Lead I, karena Lead I digunakan untuk merekam beda potensial antara tangan kanan (RA) dengan tangan kiri (LA). Hasil dari aktivitas jantung yang terekam oleh elektrokardiogram akan membentuk suatu pola yang dapat dilihat pada Gambar 2.3.



© 2008 Encyclopædia Britannica, Inc.

Gambar 2.3 Pola EKG Normal

Sumber : (Encyclopedia Britannica, 2008)

Elektrokardiogram yang memeriksa jantung normal akan menunjukkan defleksi ke atas dan ke bawah yang menampilkan kontraksi dari atrium (serambi) dan ventrikel (bilik) jantung. Defleksi ke atas pertama atau titik P merupakan kontraksi dari atrium dan biasa disebut sebagai *atrial complex*. Defleksi lainnya (Q, R, S dan T) disebabkan oleh aktivitas dari ventrikel sehingga disebut *ventricular complex*. Setiap penyimpangan dari norma dalam elektrokardiogram tertentu merupakan indikasi gangguan jantung yang mungkin terjadi.

2.2.2 Aritmia

Aritmia adalah gangguan/kelainan yang terjadi pada irama jantung. Jantung manusia merupakan organ otot yang memiliki fungsi untuk memompa darah ke seluruh tubuh. Tidak seperti tangan dan kaki yang hanya dapat bergerak setelah mendapatkan perintah dari otak, jantung memiliki pusat pengaturan sendiri sehingga secara konsisten dapat melakukan kontraksi, memompa darah ke seluruh tubuh (Fredy, 2017).

Jantung terdiri dari 2 bagian, yaitu serambi dan bilik. Serambi merupakan tempat penampungan darah pada jantung yang berada di bagian atas pada jantung. Setelah ditampung, darah akan dikirim ke bilik yang berada di bagian bawah jantung untuk selanjutnya dipompakan ke luar tubuh.



Gerakan jantung pada setiap manusia memiliki irama. Irama yang normal untuk jantung dalam satu kali berkontraksi adalah satu kali kontraksi atrium (jantung bagian atas) dan satu kali kontraksi bilik (jantung bagian bawah). Bila terjadi gangguan irama seperti dua kali kontraksi atrium kemudian satu kali kontraksi bilik atau satu kali kontraksi atrium kemudian terdapat jeda dahulu lalu terjadi kontraksi bilik, darah yang dipompa oleh jantung akan menjadi tidak optimal. Hal tersebut merupakan hal yang terjadi pada aritmia (Fredy, 2017).

Aritmia dapat dibedakan menjadi enam macam berdasarkan ritmenya, yaitu *sinus arrhythmia*, *atrial arrhythmia*, *junctional arrhythmia*, *supraventricular tachycardia (SVT) arrhythmia*, *ventricular arrhythmia* dan *AV block arrhythmia* (Clinical Education, UNM Hospitals, 2017). Aritmia ventrikular (*ventricular arrhythmia*) yang paling umum dijumpai adalah PVC (*Premature Ventricular Contraction*).

Berdasarkan polanya, PVC dibagi menjadi 2 macam yaitu *bigeminy* dan *trigeminy* (EKG Indonesia, 2015). Penjelasan dari jenis PVC aritmia tersebut dapat dilihat pada Tabel 2.2.

Tabel 2.2 Jenis PVC Aritmia

No.	Jenis PVC	Pola Grafik	Keterangan
-----	-----------	-------------	------------

1	<i>Bigeminy</i>	 <p>Sumber : (EKG Indonesia, 2015)</p>	<i>Bigeminy</i> merupakan kelainan yang terjadi dengan membentuk pola setiap <i>beat</i> setelah normal adalah PVC.
2	<i>Trigeminy</i>	 <p>Sumber : (EKG Indonesia, 2015)</p>	<i>Trigeminy</i> merupakan kelainan yang terjadi dengan membentuk pola setiap dua <i>beat</i> setelah normal adalah PVC

2.2.3 Mikrokontroler Arduino Uno

Menurut Malik & Juwana (2009) “Mikrokontroler merupakan sebuah sistem yang dibangun dengan prinsip komputer pada sebuah keping (chip) tunggal.” Hanya dengan sebuah keping IC yang dibuat maka sistem komputer dapat digunakan untuk mengontrol suatu sistem. Mikrokontroler memiliki beberapa komponen, yaitu CPU (*Central Processing Unit*), ROM (*Read Only Memory*), RAM (*Random Access Memory*) dan I/O (*Input/ Output*) dengan tambahan berupa ADC (*Analog to Digital Converter*), *Timer/ Counter* dan lain sebagainya.”

Arduino / Genuino Uno merupakan mikrokontroler dengan mikroprosesor yang memiliki tipe ATmega328P (Arduino, 2016). Mikrokontroler ini 14 digital pin *input/output* dengan 6 pin yang dapat digunakan sebagai keluaran PWM (*Pulse Width Modulation*), 6 *input* analog, *quartz crystal* 16 MHz, koneksi USB, sambungan power, *header ICSP* dan tombol reset. Semua hal tersebut digunakan untuk mendukung mikrokontroler, pengguna hanya perlu menghubungkan Arduino/ Genuino Uno ke komputer dengan menggunakan kabel serial USB dan memberikan tegangan dengan adaptor AC-DC atau baterai untuk memulai. Tampilan dari Arduino Uno dapat dilihat pada Gambar 2.4.



Gambar 2.4 Arduino Uno

Sumber: (Arduino, 2016)

Sementara itu, Arduino Uno memiliki spesifikasi yang dapat dilihat pada Tabel 2.3.

Tabel 2.3 Spesifikasi Arduino Uno

Mikrokontroler	ATmega328
Operasi tegangan	5Volt
Input tegangan	disarankan 7-11Volt
Input tegangan batas	6-20Volt
Pin I/O digital	14 (6 bisa untuk PWM)
Pin Analog	6
Arus DC tiap pin I/O	50mA
Arus DC ketika 3.3V	50mA
Memori flash	32 KB (ATmega328) dan 0,5 KB digunakan oleh bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)

Kecepatan clock	16 MHz
-----------------	--------

Sumber : (Arduino, 2016)

2.2.4 EKG Module AD8232 Arduino

Module EKG (AD8232) memiliki spesifikasi memecah sembilan sambungan dari IC sehingga dapat memilah pin, kabel, atau konektor lainnya. SDN, LO+, LO-, OUTPUT, 3.3V, GND memberikan pin penting untuk mengoperasikan monitor dengan Arduino. Module EKG AD8232 ini menyediakan pin RA (Tangan Kanan), LA (Tangan Kiri), dan RL (Kaki Kanan) untuk melampirkan dan menggunakan sensor lainnya. Selain itu, module EKG AD8232 memiliki lampu indikator LED yang akan berdenyut sesuai dengan ritme detak jantung. Tampilan sensor EKG AD8232 ini dapat dilihat pada Gambar 2.5.



Gambar 2.5 Sensor EKG AD8232

Sumber: (Sparkfun, 2014)

Sensor EKG Module AD8232 Arduino memiliki enam pin dengan deskripsi yang dapat dilihat pada Tabel 2.4.

Tabel 2.4 Deskripsi Pin Sensor EKG Module AD8232

Pin	Simbol	Fungsi
1	GND	Ground (0V)
2	Vcc	Power Supply (3,3V)
3	Output	Mengirim data keluaran berupa analog ke pin yang diinginkan.
4	LO-	Merupakan input negatif dari elektroda.
5	LO+	Merupakan input positif dari elektroda.
6	SDN	Melakukan <i>shutdown</i> .

2.2.5 Naïve Bayes Algorithm

Algoritma *Naïve Bayes* adalah metode klasifikasi yang menggunakan probabilitas dan statistik yg ditemukan oleh ilmuwan Inggris Thomas Bayes. Algoritma *Naïve Bayes* melakukan prediksi peluang depan berdasarkan pengalaman sebelumnya sehingga dikenal sebagai Teorema Bayes. Ciri utama dari *Naïve Bayes Classifier* ini adalah asumsi yang kuat (naïf) akan independensi dari masing-masing kondisi/kejadian.

Naïve Bayes memiliki hasil kesimpulan yang sangat baik dibanding dengan model klasifikasi lainnya. Hal ini dibuktikan pada jurnal Xhemali, Daniela, Chris J. Hinde, and Roger G. Stone. "*Naïve Bayes vs. decision trees vs. neural networks in the classification of training web pages.*" (2009), mengungkapkan bahwa "Klasifikasi *Naïve Bayes* memiliki tingkat akurasi yg lebih baik dibandingkan dengan model klasifikasi lainnya".

Keuntungan menggunakan metode *Naïve Bayes* adalah metode ini hanya membutuhkan jumlah data latih (training data) yang sedikit untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasiannya. Karena yang diasumsikan sebagai variabel *independent*, maka hanya *varian* dari suatu variabel dalam sebuah kelas yang dibutuhkan untuk menentukan klasifikasi, bukan keseluruhan dari matriks kovarians.

Berikut merupakan rumus dari teorema Bayes (Baber, 2010) :

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \quad (2.1)$$

Keterangan dari **Persamaan (2.1)** yaitu:

$(y|x)$: Peluang *posterior* (probabilitas kondisional) dari suatu hipotesis kelas y yang terjadi setelah diberikan data x .

$(x|y)$: Peluang *likelihood* dari sebuah data x yang terjadi akan mempengaruhi hipotesis kelas y .

(y) : Peluang *prior* (awal) hipotesis kelas y yang terjadi tanpa memperhatikan data yang diberikan.

(x) : Peluang *evidence* x yang terjadi tanpa memperhatikan hipotesis kelas lainnya, yaitu jumlah total dari semua peluang *likelihood* yang dikalikan dengan peluang *prior*.

Hipotesis yang digunakan dalam teorema Bayes merupakan label kelas yang menjadi target dalam sebuah klasifikasi, sedangkan *evidence* merupakan fitur yang akan menjadi nilai masukan dalam klasifikasi. *Naïve Bayes* dilambangkan dengan simbol $(X|Y)$, dimana X merupakan masukan berupa nilai fitur dan Y merupakan kelas dalam sebuah klasifikasi. Notasi $(X|Y)$ berarti peluang kelas Y didapatkan setelah fitur-fitur X diamati, notasi ini merupakan peluang *likelihood* dan (Y) merupakan notasi dari peluang prior. Berikut ini merupakan persamaan untuk rumus *Naïve Bayes* (Baber, 2010) :

$$P(Y|X) = \frac{P(Y) \prod_{i=1}^a P(X_i|Y)}{P(X)} \quad (2.2)$$

Keterangan dari **Persamaan (2.2)** yaitu:

$(Y|X)$: Peluang *posterior* (probabilitas kondisional) dari suatu kelas Y akan terjadi setelah mengamati fitur-fitur X.

$\prod_{i=1}^a P(X_i|Y)$: Peluang *likelihood* dari masing-masing fitur X terjadi akan mempengaruhi kelas Y.

(Y) : Peluang *prior* (awal) hipotesis kelas y terjadi tanpa memperhatikan fitur yang diberikan.

(X) : Peluang *evidence* X terjadi tanpa memperhatikan kelas/evidence lainnya, yakni jumlah total dari semua peluang likelihood yang dikalikan dengan peluang prior.

Perhitungan klasifikasi untuk setiap kelas Y yang berbeda akan memiliki nilai fitur (X) yang sama, sehingga dalam penentuan klasifikasi *Naïve Bayes* selanjutnya ditentukan dari nilai peluang terbesar pada setiap kelas Y dari hasil perhitungan $(Y) \prod_{i=1}^q P(X_i|Y)$ (Astuti, 2016).

Terdapat beberapa permasalahan untuk menentukan nilai peluang dari suatu kondisi, yaitu dengan menghitung peluang dari data diskrit. Namun dalam kenyataannya tidak semua data memiliki bentuk diskrit, terdapat data lain yang memiliki bentuk data kontinyu. Untuk itu dalam melakukan proses klasifikasi pada data kontinyu dengan *Naïve Bayes* terdapat 2 cara yaitu (Astuti, 2016) :

1. Melakukan proses perubahan data kontinyu menjadi data diskrit (diskritisasi) terhadap setiap fitur yang akan diestimasi.
2. Menganggap setiap fitur sesuai dengan data latih menggunakan fungsi *univariate normal (Gaussian)* distribution yang ditunjukkan pada **Persamaan (2.3)**, dimana parameter utama dari fungsi *Gaussian* ini adalah mean (μ) dan varian (σ^2).

$$P(X = x_i|Y = y_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}} \quad (2.3)$$

Parameter μ_{ij} bisa didapatkan dari nilai rata-rata (*mean*) pada sampel X_i (\bar{x}) dari semua data latih yang menjadi milik kelas y_i , sedangkan σ_{ij}^2 dapat diperkirakan dari varian sampel (s^2) dari data latih.

Adapun fungsi untuk mencari nilai rata-rata (*mean*) dapat dilihat pada **Persamaan (2.4)**.

$$\bar{x} = \frac{\sum_{i=1}^n X_i}{n} \quad (2.4)$$

Perhitungan nilai rata-rata (*mean*) dilakukan dengan menjumlahkan seluruh nilai data pada kelompok sampel, kemudian dibagi dengan total sampel tersebut. Dimana \bar{x} merupakan rata-rata hitung, x_i merupakan nilai sampel ke-i, dan n merupakan jumlah sampel.

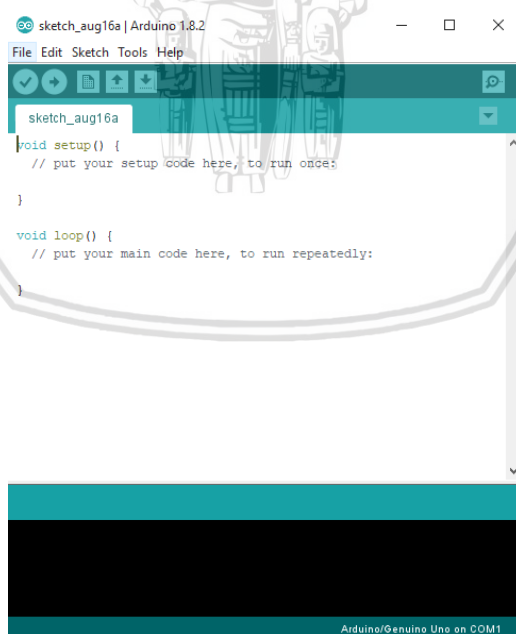
Kemudian fungsi untuk mencari nilai standar deviasi dapat dilihat pada **Persamaan (2.5)**.

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \quad (2.5)$$

Untuk menghitung standar deviasi yaitu dengan mengurangi setiap nilai data dengan rata-rata kelompok dari data tersebut, selanjutnya semua hasil dijumlahkan yang kemudian dibagi dengan jumlah data secara keseluruhan dikurangkan 1, hasilnya di akarkan. Dimana s merupakan standar deviasi (simpangan baku), x_i merupakan nilai x ke i , \bar{x} merupakan rata-rata dan n adalah ukuran sampel.

2.2.6 Pemrograman C Arduino

Pemrograman merupakan suatu cara untuk memasukan perintah-perintah yang ingin dijalankan pada mikrokontroler dengan bahasa yang dimengerti oleh komputer. Terdapat macam-macam bahasa pemrograman komputer, seperti Java, C#, C++, C dan lain sebagainya. Bahasa pemrograman Arduino merupakan bahasa pemrograman C/ C++ yang dapat dipanggil kemudian di-*compile* dan di-*upload* (Arduino, 2016). Pemograman ini merupakan pemograman tingkat rendah, yang artinya pemograman yang dilakukan langsung dilakukan ke hardware. Oleh sebab itu, pemograman Arduino langsung ditujukan kepada pin yang digunakan kepada *source code*. Semua bahasa C dan C++ yang dibangun harus sesuai dengan standar avrg++ agar dapat mendukung dan bekerja di Arduino. Untuk melakukan pemrograman C arduino sendiri dapat menggunakan IDE Arduino yang dapat dilihat pada gambar 2.6.



Gambar 2.6 Arduino IDE

2.2.7 Pemrograman Java Processing

Processing merupakan perangkat lunak (software) dan bahasa yang digunakan untuk mempelajari bagaimana melakukan kode program dengan fungsi

menampilkan suatu seni visual (Processing, 2014). Perangkat lunak Processing ini memiliki beberapa bahasa pemrograman yang dapat digunakan, namun secara default Processing menggunakan bahasa pemrograman Java. Bahasa pemrograman Java cocok digunakan untuk perangkat lunak Processing karena bahasa pemrograman Java merupakan pemrograman yang berorientasi objek dan dapat dijalankan untuk segala macam jenis sistem operasi. Untuk melakukan pemrograman Java ini dapat menggunakan IDE Processing yang dapat dilihat pada Gambar 2.7.



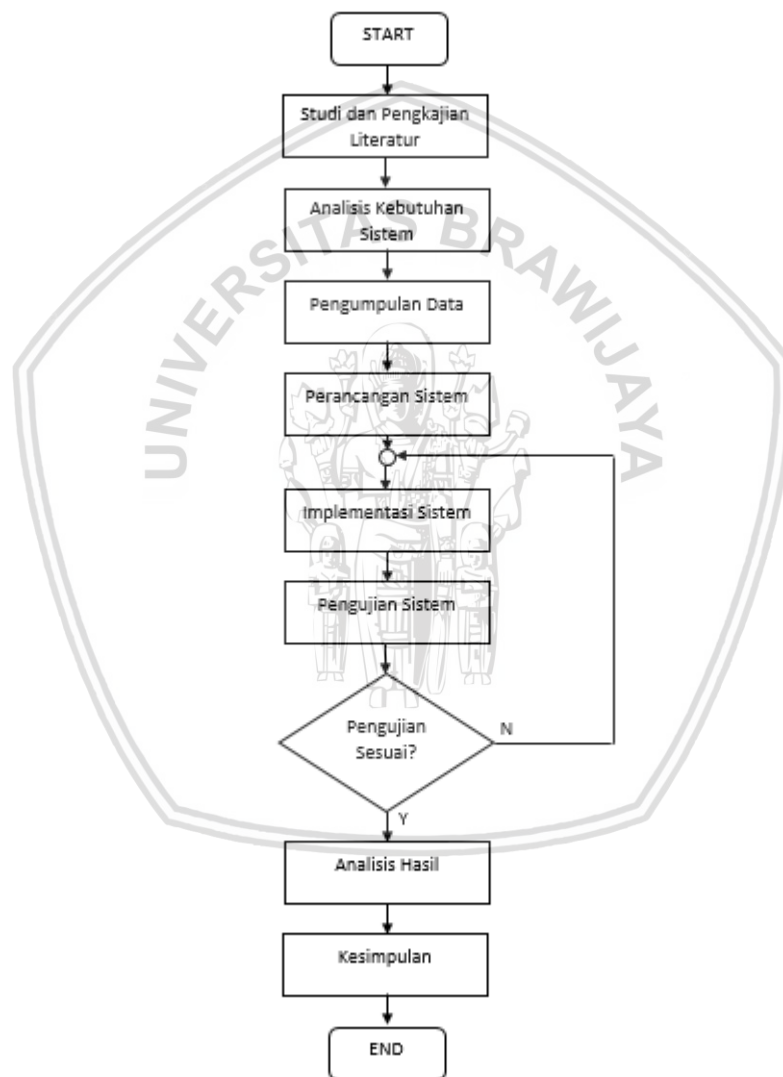
Gambar 2.7 Perangkat Lunak Processing

BAB 3 METODE PENELITIAN

Bab ini menjelaskan mengenai langkah-langkah yang dilaksanakan dalam penulisan yang meliputi studi dan pengkajian literatur, analisis kebutuhan sistem, perancangan sistem, implementasi sistem, pengujian dan analisis, dan kesimpulan.

3.1 Alur Metodologi Penelitian

Secara umum, alur metodologi yang dilakukan pada penelitian ini dapat dilihat pada diagram alur Gambar 3.1.



Gambar 3.1 Diagram Alir Metode Penelitian

Berdasarkan diagram alir pada Gambar 3.1 dapat dilihat semua proses yang dilakukan secara berurutan yang dimulai dari studi dan pengkajian literatur hingga kesimpulan dan saran. Beberapa tahapan memiliki kondisi atau syarat agar dapat lanjut ke tahapan selanjutnya.

3.2 Studi Pengkajian Literatur

Perancangan dan implementasi pada penelitian ini, perlu adanya studi literatur. Literatur digunakan sebagai teori penguat dan landasan dasar yang digunakan dalam penelitian. Teori pendukung tersebut dapat diambil dari buku, jurnal, paper dan internet. Beberapa literatur yang digunakan pada penelitian ini antara lain sebagai berikut :

1. Elektrokardiografi

Mempelajari terkait pengertian dari EKG, cara mengukur, teknik penempatan elektroda yang digunakan dan pola dari gelombang jantung.

2. Aritmia

Mempelajari terkait pengertian dari jantung aritmia, penyebab dan pola dari gelombang EKG aritmia.

3. Mikrokontroler Arduino Uno

Mempelajari terkait konfigurasi penggunaan Arduino Uno sebagai tempat untuk melakukan segala proses data baik dari konfigurasi pin perangkat keras maupun segi penggunaan logika dan algoritma yang sesuai.

4. Sensor EKG Module AD8232 Arduino

Mempelajari terkait konfigurasi, datasheet dan spesifikasi dari sensor EKG Module AD8232 Arduino.

5. *Naïve Bayes*

Mempelajari mengenai pengertian klasifikasi *Naïve Bayes*, kekurangan serta kelebihan dari metode ini dan rumus yang diterapkan dalam melakukan klasifikasi.

3.3 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem bertujuan untuk menganalisis semua kebutuhan yang diperlukan oleh sistem yang akan dibangun. Kebutuhan yang diperlukan oleh sistem dalam penelitian ini adalah sebagai berikut :

1. Sensor AD8232 dapat membaca nilai tegangan pada jantung yang dideteksi dengan satuan mV.
2. Data dari sensor AD8232 dapat diolah secara *real time*, sehingga sistem dapat menentukan hasil klasifikasi jenis kelainan detak jantung Premature Ventricular Contraction Aritmia berdasarkan metode *Naïve Bayes*.
3. Sistem dapat menampilkan hasil keluaran berupa grafik EKG, nilai dari R interval, nilai bpm yang didapatkan dan hasil klasifikasi dari metode *Naïve Bayes* pada software aplikasi Processing.

Analisis kebutuhan dari sistem yang akan dibangun terbagi menjadi dua kebutuhan yaitu kebutuhan perangkat lunak dan kebutuhan perangkat keras.

3.3.1 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan pada sistem ini adalah Arduino IDE dengan versi 1.8.5, arduino library, dan Processing IDE yang digunakan untuk menampilkan keluaran pada sistem.

3.3.2 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang digunakan untuk membuat sistem ini adalah sebagai berikut :

1. Mikrokontroler Arduino Uno R3 Atmega328P
2. Sensor Arduino Kit AD8232 Heart Beat Rate
3. Kabel Jumper
4. Laptop

3.4 Pengumpulan Data

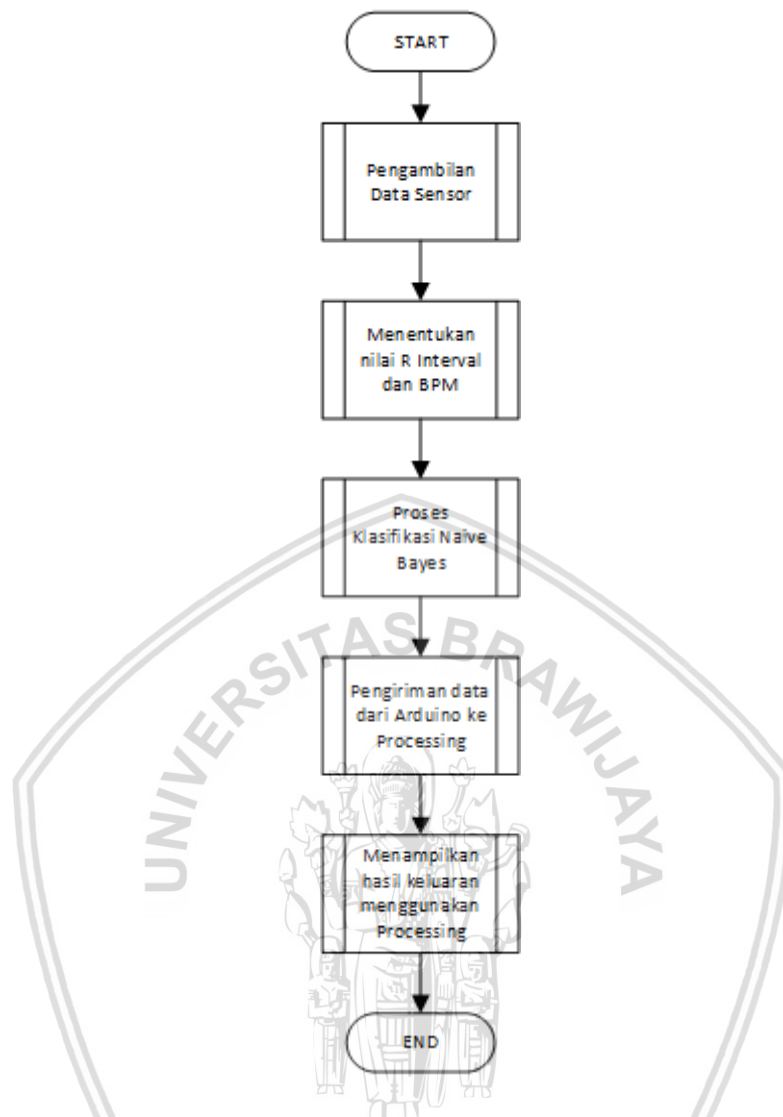
Proses pengumpulan data yang dilakukan untuk mengumpulkan fitur data set sistem berupa waktu R interval dan nilai bpm dari data pasien yang telah direkam di MIT-BIH *Arrhythmia Database*. Waktu pelaksanaan untuk melakukan pengambilan data dilakukan pada bulan Januari 2018. Data set yang didapatkan akan dijadikan data latih sebagai basis pengetahuan Sistem Pendeteksi *Premature Ventricular Contraction* Aritmia untuk mengukur tingkat akurasi sistem. Pengambilan sampel dilakukan dengan menggunakan teknik *judgement sampling*, yaitu teknik pengambilan sampel dengan berdasarkan karakteristik yang disesuaikan dengan maksud dan tujuan tertentu.

3.5 Perancangan Sistem

Perancangan sistem dilakukan setelah analisis kebutuhan sistem telah terpenuhi. Tujuan dilakukan perancangan sistem agar pada implementasinya berjalan secara sistematis. Perancangan sistem terbagi menjadi dua bagian yaitu perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*).

2.3.1 Perancangan Perangkat Lunak

Perancangan perangkat lunak merupakan rancangan sistem yang dibuat berdasarkan dari kebutuhan perangkat lunak. Rancangan dari perangkat lunak sistem secara umum dapat dilihat pada Gambar 3.2.

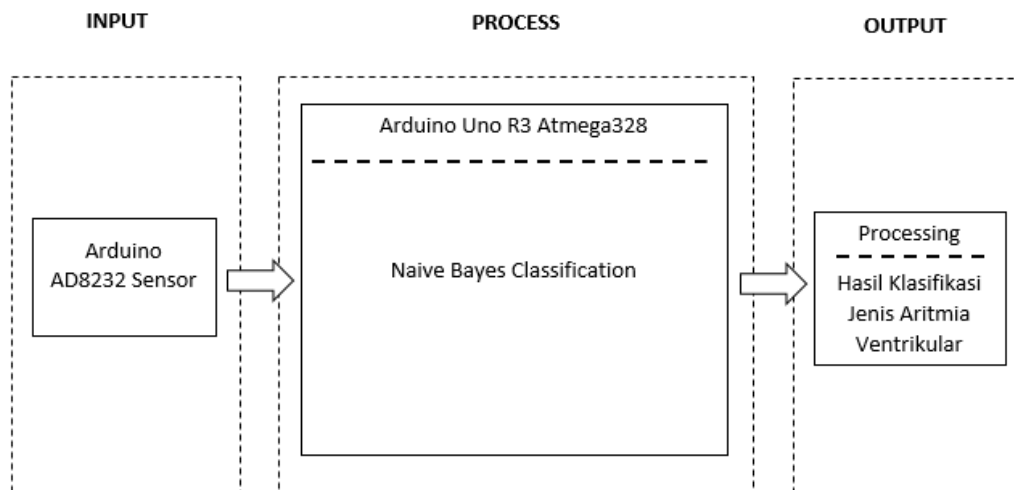


Gambar 3.2 Diagram Alir Perancangan Perangkat Lunak

Pada Gambar 3.2 menerangkan bahwa sistem akan dapat bekerja apabila terdapat pembacaan data dari sensor AD8232 yang kemudian akan dilakukan perhitungan peluang untuk penentuan hasil klasifikasi jenis *Premature Ventricular Contraction* Aritmia sesuai dengan metode *Naïve Bayes*.

2.3.2 Perancangan Perangkat Keras

Perancangan perangkat keras merupakan rancangan sistem yang dibuat berdasarkan dari kebutuhan perangkat keras. Gambaran umum dari rancangan perangkat keras sistem ini dapat digambarkan menggunakan blok diagram yang dapat dilihat pada Gambar 3.3.



Gambar 3.3 Blok Diagram Sistem

Pembacaan dari sensor AD8232 EKG Module Arduino merupakan masukan dari sistem. Kemudian mikrokontroller Arduino Uno akan membaca masukan tersebut dan melakukan proses komputasi. Dalam prosesnya, sistem menggunakan metode *Naïve Bayes* untuk menarik kesimpulan untuk hasil keluaran yang akan ditampilkan di Processing.

3.6 Implementasi Sistem

Implementasi system pada penelitian ini dilakukan dengan berdasarkan pada perancangan sistem yang sudah dibuat sebelumnya. Pada implementasi sistem juga mencamtumkan gambaran sistem yang telah dibuat serta potongan kode pemrograman yang digunakan pada sistem.

3.7 Pengujian dan Analisis

Pada tahap ini akan dilakukan pengujian dan analisis dari hasil pengujian yang dilaksanakan sesuai dengan parameter pada perancangan sistem sebelumnya. Terdapat beberapa skenario yang akan dilakukan pada pengujian sistem ini, yaitu :

1. Pengujian pada sensor EKG Arduino AD8232 dalam menghitung bpm pada seseorang.
2. Pengujian akurasi pada hasil klasifikasi metode *Naïve Bayes*.
3. Pengujian waktu komputasi pemrosesan pada sistem.

3.8 Kesimpulan dan Saran

Kesimpulan dari sistem didapatkan setelah proses pengumpulan data, perancangan, implementasi, pengujian dan analisis telah berhasil dilakukan. Kesimpulan ini diambil berdasarkan hasil dari pengujian dan analisis yang merupakan penerapan dari algoritma yang digunakan terhadap perancangan. Diharapkan isi dari kesimpulan dapat menjadi acuan agar sistem aplikasi ini dapat

dikembangkan lebih lanjut. Selain itu, terdapat saran pada akhir penulisan yang bertujuan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan untuk pengembangan sistem selanjutnya.



BAB 4 REKAYASA KEBUTUHAN

Pada bab ini dijelaskan secara rinci mengenai gambaran umum sistem, analisis kebutuhan fungsional dan non-fungsional, kebutuhan perangkat keras, kebutuhan perangkat lunak dan batasan desain sistem.

4.1 Gambaran Umum Sistem

Sistem Pendeteksi Premature Ventricular Contraction Aritmia Menggunakan Metode *Naïve Bayes* merupakan sistem yang dapat menentukan apakah seseorang mengalami Premature Ventricular Contraction Aritmia atau tidak dengan parameter berupa nilai interval tiap detak jantung dan jumlah bpm. Nilai interval dan bpm tersebut didapat dari sensor EKG Arduino AD8232. Setelah nilai dari kedua parameter tersebut didapatkan, dilakukan klasifikasi dengan menggunakan metode *Naïve Bayes*. Sistem ini digunakan untuk membantu seseorang mengetahui apakah orang tersebut mengalami gejala Premature Ventricular Contraction Aritmia atau tidak, dimana pada saat ini untuk mendeteksi penyakit jantung aritmia hanya dapat dilakukan di rumah sakit dengan biaya yang relatif mahal. Penggunaan metode *Naïve Bayes* pada sistem ini dikarenakan dalam melakukan klasifikasi, sudah diketahui terlebih dahulu jenis klasifikasi yang akan ditentukan yaitu kondisi normal, bigeminy dan trigeminy. Selain itu metode klasifikasi *Naïve Bayes* dapat menghasilkan tingkat akurasi yang tinggi berdasarkan dengan jumlah peluang yang dianggap benar berdasarkan data latih sebelumnya. Semakin banyak jumlah data latih yang benar, maka semakin tinggi tingkat keakuratan pada sistem.

4.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk mengetahui dan memperkirakan semua kebutuhan yang dibutuhkan oleh Sistem Pendeteksi Premature Ventricular Contraction Aritmia ini. Analisis kebutuhan sistem ini dapat dijabarkan menjadi dua yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional ini dilakukan berdasarkan data dan informasi yang harus dimiliki sistem sehingga sistem dapat memenuhi tujuan dan sesuai dengan yang diharapkan. Kebutuhan fungsional dari sistem aplikasi memiliki persyaratan sebagai berikut :

1. Sensor EKG AD8232 dapat membaca nilai tegangan dari jantung
Sensor EKG AD8232 harus dapat membaca nilai berupa tegangan dari detak jantung manusia serta dapat menghitung nilai bpmnya.
2. Arduino Uno dapat mengolah data masukan dari sensor AD8232
Data dari sensor EKG AD8232 harus dapat diolah untuk menghasilkan penentuan klasifikasi jenis PVC aritmia menggunakan metode *Naïve Bayes*.

3. Nilai fitur dapat dijadikan acuan untuk melakukan klasifikasi jenis PVC Aritmia menggunakan metode *Naive Bayes*
Nilai dari R Interval dan BPM yang merupakan fitur klasifikasi metode *Naive Bayes* pada sistem ini yang didapatkan dari proses perhitungan pada data sensor EKG AD8232 dapat dijadikan sebagai nilai *input* yang digunakan dengan membandingkan nilai tersebut dengan nilai yang ada pada data latih, kemudian sistem dapat melakukan klasifikasi menjadi tiga jenis kelas yaitu normal, *bigeminy* dan *trigeminy*.
4. Sistem dapat menampilkan hasil keluaran
Sistem harus dapat menampilkan nilai keluaran sensor AD8232 berupa grafik EKG dan dapat menampilkan nilai dari R interval, perhitungan bpm serta hasil klasifikasi menggunakan tampilan antarmuka pada *software* Processing.

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional ini digunakan untuk mengetahui kebutuhan sistem aplikasi yang berhubungan dengan perangkat keras ataupun perangkat lunak. Kebutuhan perangkat keras dari Sistem Pendeteksi Premature Ventricular Contraction Aritmia ini antara lain sebagai berikut :

1. Mikrokontroler Arduino Uno R3
Mikrokontroler Arduino Uno R3 berfungsi sebagai otak pengolahan informasi dan pemrosesan perancangan pada sistem.
2. Sensor EKG AD8232 Arduino dan Elektroda
Sensor ini digunakan untuk membaca nilai detak jantung manusia dengan menggunakan elektroda sebagai perantaranya.
3. Kabel Jumper
Dibutuhkan untuk menghubungkan satu perangkat dengan perangkat lainnya.
5. Laptop Asus A46C
Sebagai sumber daya dari sistem dan sebagai media untuk membuat/menjalankan program dari sistem.

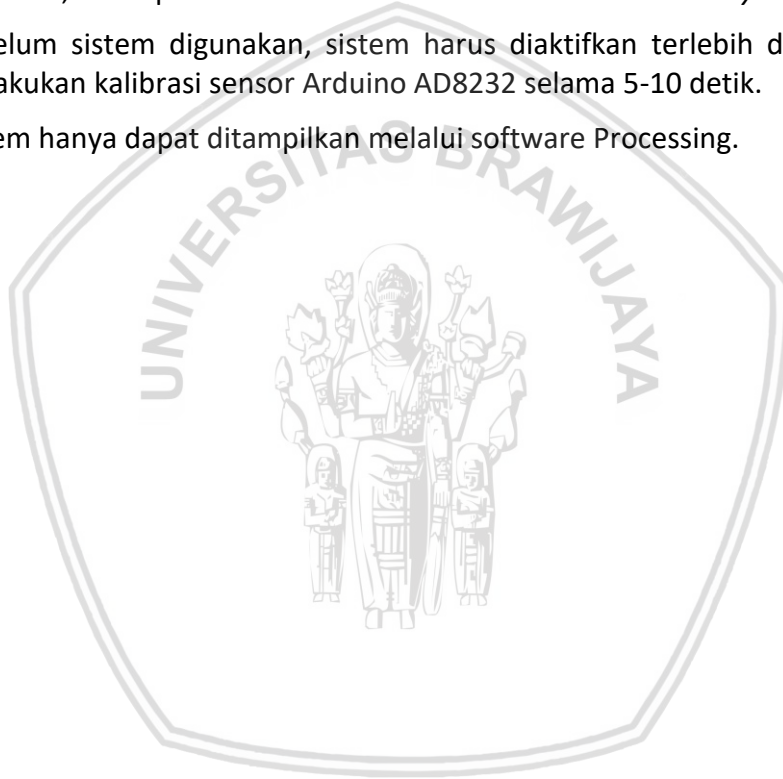
Sementara kebutuhan perangkat lunak dari Sistem Pendeteksi Premature Ventricular Contraction Aritmia ini adalah sebagai berikut :

1. Arduino IDE v.1.8.5
Perangkat lunak yang digunakan untuk mengoperasikan mikrokontroler arduino uno.
2. Arduino *library* math.h dan fir.h
Suatu fungsi pada Arduino IDE yang digunakan untuk melakukan perhitungan matematika yang rumit dan melakukan filter pada EKG.
3. Processing v.2.2.1
Perangkat lunak yang digunakan untuk menampilkan keluaran grafik sinyal EKG, nilai delapan R interval, nilai BPM dan hasil klasifikasi.

4.3 Batasan Desain Sistem

Dalam pembuatan Sistem Pendeteksi Penyakit Jantung aritmia ini memiliki beberapa batasan sehingga lingkup pembahasan, perancangan dan implementasinya dibatasi sedemikian rupa. Adapun batasan-batasan dari sistem ini adalah sebagai berikut :

1. Sistem hanya mengklasifikasi tiga jenis kondisi, yaitu normal, bigeminy dan trigeminy.
2. Sistem hanya dapat melakukan pendeteksian dan pengklasifikasian ketika elektroda EKG telah dipasang.
3. Sistem hanya memberikan keluaran berupa tujuh nilai interval setiap jantung berdetak, nilai bpm dan hasil klasifikasi dari metode *Naïve Bayes*.
4. Sebelum sistem digunakan, sistem harus diaktifkan terlebih dahulu untuk melakukan kalibrasi sensor Arduino AD8232 selama 5-10 detik.
5. Sistem hanya dapat ditampilkan melalui software Processing.



BAB 5 PERANCANGAN DAN IMPLEMENTASI

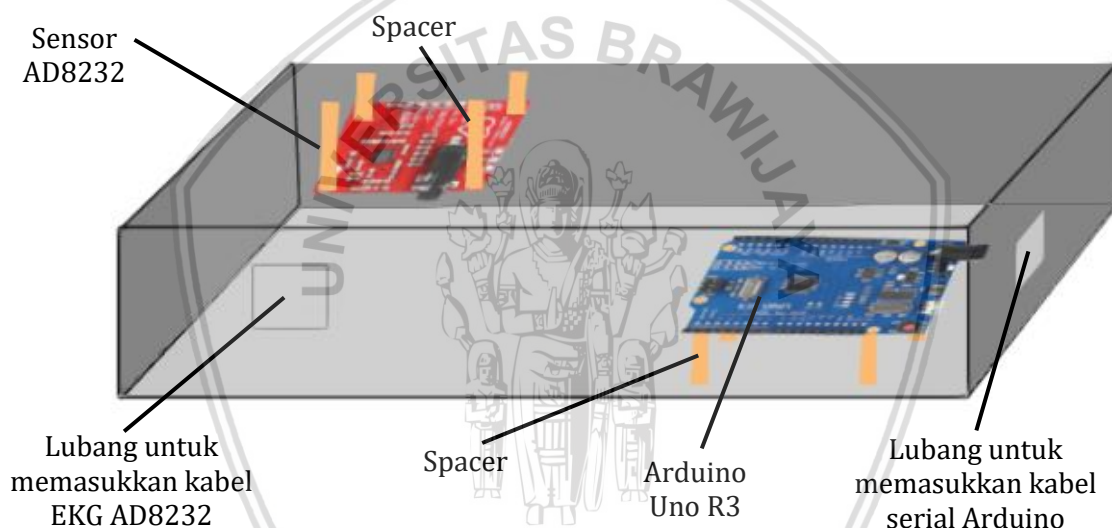
Bab ini menerangkan proses perancangan dan implementasi dari sistem secara terperinci pada sisi perangkat lunak maupun perangkat keras.

5.1 Perancangan Sistem

Pada perancangan sistem ini akan dijelaskan mengenai pembuatan perancangan sistem dimulai dari perancangan prototype alat pada sistem, perancangan perangkat keras pada sistem dan perancangan perangkat lunak pada sistem.

5.1.1 Perancangan *Prototype* Sistem

Tahap perancangan Prototype pada Sistem Pendeteksi Premature Ventricular Contraction Aritmia ini dapat diilustrasikan pada Gambar 5.1.



Gambar 5.1 Desain *Prototype* Sistem

Dari Gambar 5.1 dapat dilihat bahwa desain dari Sistem Pendeteksi Premature Ventricular Contraction Aritmia ini dibungkus dengan kotak hitam yang berukuran $15 \times 10 \times 5 \text{ cm}^3$. Pada bagian samping dan depan dari kotak hitam tersebut terdapat lubang yang digunakan sebagai ruang masuknya kabel serial Arduino dan kabel EKG sensor AD8232. Kemudian, Arduino Uno dan Sensor AD8232 dipasangkan dengan kotak hitam menggunakan spacer berukuran 2 cm yang membuat kedua alat tersebut tidak berpindah tempat. Arduino Uno terletak pada bagian bawah kotak hitam, sementara sensor AD8232 terletak pada bagian atas kotak hitam.

5.1.2 Perancangan Perangkat Keras Sistem

Perancangan perangkat keras Alat Pendeteksi *Premature Ventricular Contraction* Aritmia ini dibagi menjadi dua bahasan, yaitu perancangan perangkat keras mikrokontroler dan sensor dan perancangan perangkat keras penempatan elektroda.

Nomor Pin	Keterangan
2	GND
3	3,3V
4	OUTPUT
5	LO-
6	LO+

Berdasarkan Tabel 5.1 dapat dilihat pin pada sensor AD8232 yang digunakan hanyalah pin 2 hingga pin 6. Kemudian untuk melakukan koneksi antar pin pada sensor AD8232 dan Arduino Uno dapat dijelaskan pada Tabel 5.2.

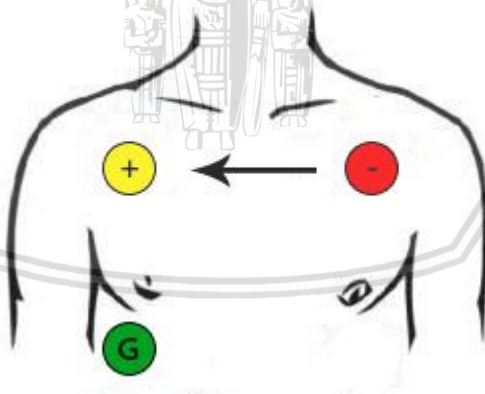
Tabel 5.2 Keterangan Koneksi Pin Sensor AD8232 Dengan Arduino Uno

Pin Sensor AD8232	Pin Arduino Uno
GND	GND
3,3V	3,3V
OUTPUT	A0
LO-	11
LO+	10

Koneksi pin antara sensor AD8232 dengan pin Arduino Uno pada sistem ini dihubungkan langsung menggunakan *jumper male-female*, dimana ujung *male* digunakan untuk menghubungkan ke Arduino Uno dan ujung *female* digunakan untuk menghubungkan ke sensor AD8232.

5.1.2.2 Perancangan Penempatan Elektroda

Perancangan perangkat keras penempatan elektroda pada sistem ini dilakukan berdasarkan teori penempatan elektroda EKG menggunakan 3 Lead yang dapat dilihat pada Gambar 5.3.



Gambar 5.3 Penempatan Elektroda pada Sistem Pendeteksi *Premature Ventricular Contraction* Aritmia

Pada Gambar 5.3 dijelaskan bahwa pada Sistem Pendeteksi *Premature Ventricular Contraction* Aritmia ini terdapat 3 elektroda yang digunakan, yaitu elektroda berwarna kuning sebagai kutub positif, elektroda berwarna merah sebagai kutub negatif dan elektroda berwarna hijau sebagai ground. Penempatan dari masing-masing elektroda tersebut dapat dilihat pada Tabel 5.3.

Tabel 5.3 Penempatan Elektroda

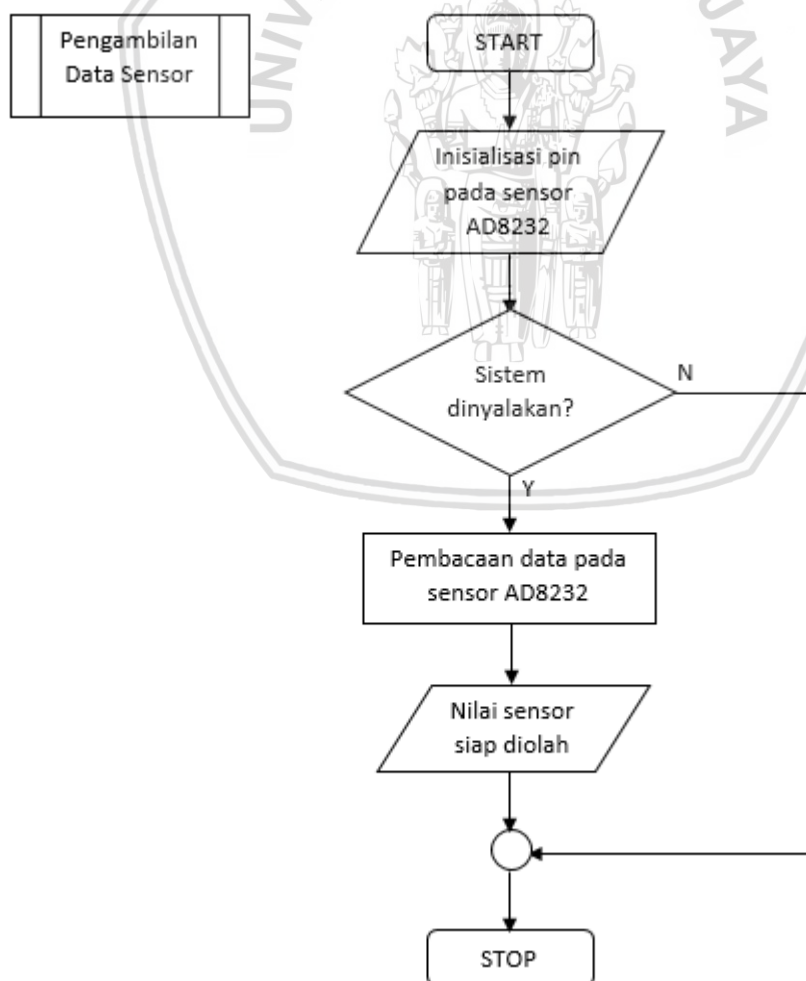
Warna Elektroda	Tempat
Kuning	RA (Dada atas bagian kanan)
Merah	LA (Dada atas bagian kiri)
Hijau	RL (Perut bagian kanan)

5.1.3 Perancangan Perangkat Lunak Sistem

Perancangan perangkat lunak sistem ini dibagi menjadi tiga bahasan, yaitu perancangan perangkat lunak pada mikrokontroler untuk pengambilan data dari sensor yang ingin diolah, perancangan perangkat lunak pada mikrokontroler untuk menentukan nilai R interval dan BPM dan perancangan perangkat lunak untuk melakukan proses klasifikasi dengan menggunakan metode *Naïve Bayes*.

5.1.3.1 Perancangan Pengambilan Data Sensor

Proses perancangan pengambilan data sensor pada Sistem Pendeteksi Premature Ventricular Contraction Aritmia ini dapat digambarkan melalui diagram alir pengambilan data sensor pada Gambar 5.4.



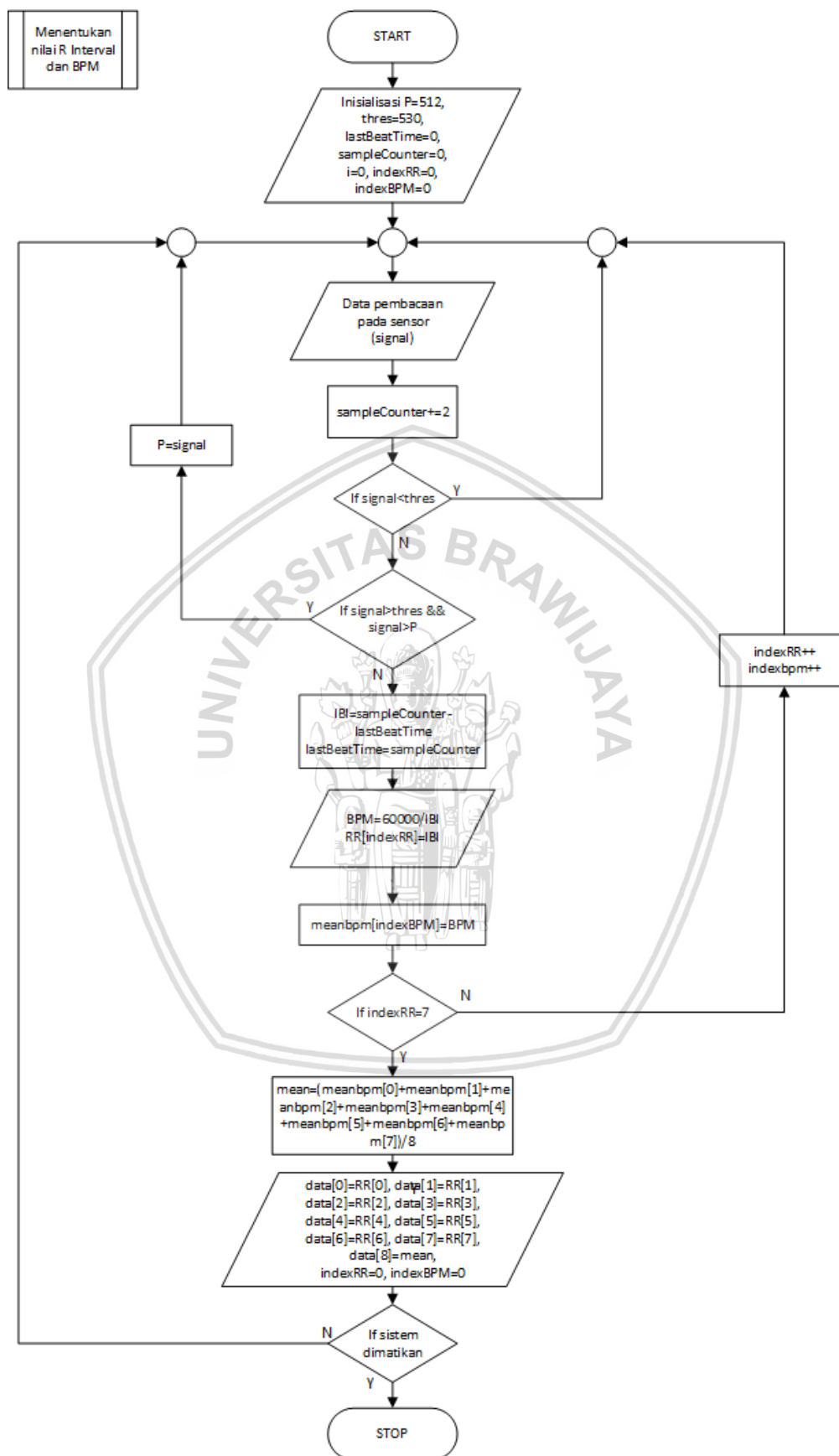
Gambar 5.4 Diagram Alir Perancangan Perangkat Lunak Pengambilan Data Sensor

Diagram alir pada Gambar 5.4 menjelaskan bahwa untuk menggunakan sensor AD8232, hal yang pertama kali dilakukan adalah melakukan inisialisasi pin sensor AD8232 dengan mikrokontroler arduino uno. Setelah terinisialisasi, sistem akan membaca nilai yang didapat dari sensor AD8232, kemudian Arduino Uno akan mengambil dan mengolah lebih lanjut nilai tersebut.

5.1.3.2 Perancangan Penentuan Nilai R Interval dan BPM

Setelah nilai sensor didapatkan, maka tahap berikutnya adalah mencari fitur dari klasifikasi *Naïve Bayes* yang diinginkan yaitu nilai R interval dan nilai BPM. Untuk proses perancangan penentuan nilai R interval dan BPM pada sistem ini dapat digambarkan menggunakan diagram alir yang dapat dilihat pada Gambar 5.5.





Gambar 5.5 Diagram Alir Perancangan Penentuan Nilai R Interval dan BPM

Dari diagram alir perancangan penentuan nilai R interval dan BPM pada Gambar 5.5 di atas menjelaskan bahwa setiap sistem melakukan pembacaan nilai pada sensor, sistem akan melakukan increment dua yang berarti sistem akan membaca data pada sensor setiap 2 milisekon. Setelah itu, sistem akan memasuki tahap kondisi. Jika nilai dari variabel signal melebihi nilai thres maka akan masuk ke kondisi selanjutnya, jika tidak maka sistem akan kembali membaca nilai pada sensor. Pada kondisi selanjutnya, jika nilai signal melebihi nilai thres yang merupakan nilai batas dan nilai signal melebihi P yang dijadikan sebagai nilai puncak, maka nilai Signal menjadi nilai P. Kondisi ini dilakukan sampai nilai P menjadi nilai tertinggi atau sebagai titik puncak. Jika kondisi tersebut bernilai false, maka mendefinisikan nilai IBI (Inter-Beat Interval) adalah nilai dari sampleCounter dikurangi nilai lastBeatTime, kemudian nilai lastBeatTime akan menjadi nilai sampleCounter. Setelah nilai IBI berhasil didapatkan, sistem akan melakukan perhitungan untuk mencari nilai BPM berdasarkan **Persamaan (5.1)**.

$$[BPM] = \frac{60000}{IBI} \quad (5.1)$$

Nilai BPM pada sistem ini didapatkan dari hasil perhitungan 60000 yang merupakan satu menit dalam milisekon dibagi dengan nilai IBI yang merupakan waktu ketika jantung berdetak pada beat tersebut. Hasil dari nilai BPM akan dibulatkan ke nilai terdekat sehingga nilai dari BPM tidak akan koma. Selain itu mendefinisikan nilai dari RR[indexRR] yang merupakan nilai R interval, dimana nilai R interval merupakan nilai IBI sebelumnya. Kemudian, nilai meanbpm dengan indeks meanBPM sama dengan nilai BPM, dimana nilai meanBPM ini akan digunakan untuk menghitung nilai BPM yang akan dijadikan fitur klasifikasi *Naïve Bayes*. Masuk ke dalam kondisi jika index RR bernilai tujuh adalah false, maka sistem akan melakukan inkremen kepada indexRR dan index BPM kemudian kembali membaca nilai sensor. Jika kondisi bernilai true, maka akan masuk ke dalam perhitungan dari mean dimana nilai mean didapatkan dari **Persamaan (5.2)**.

$$[mean] = \frac{\sum_{i=0}^7 meanbpm_i}{8} \quad (5.2)$$

Keterangan dari **Persamaan (5.2)** yaitu :

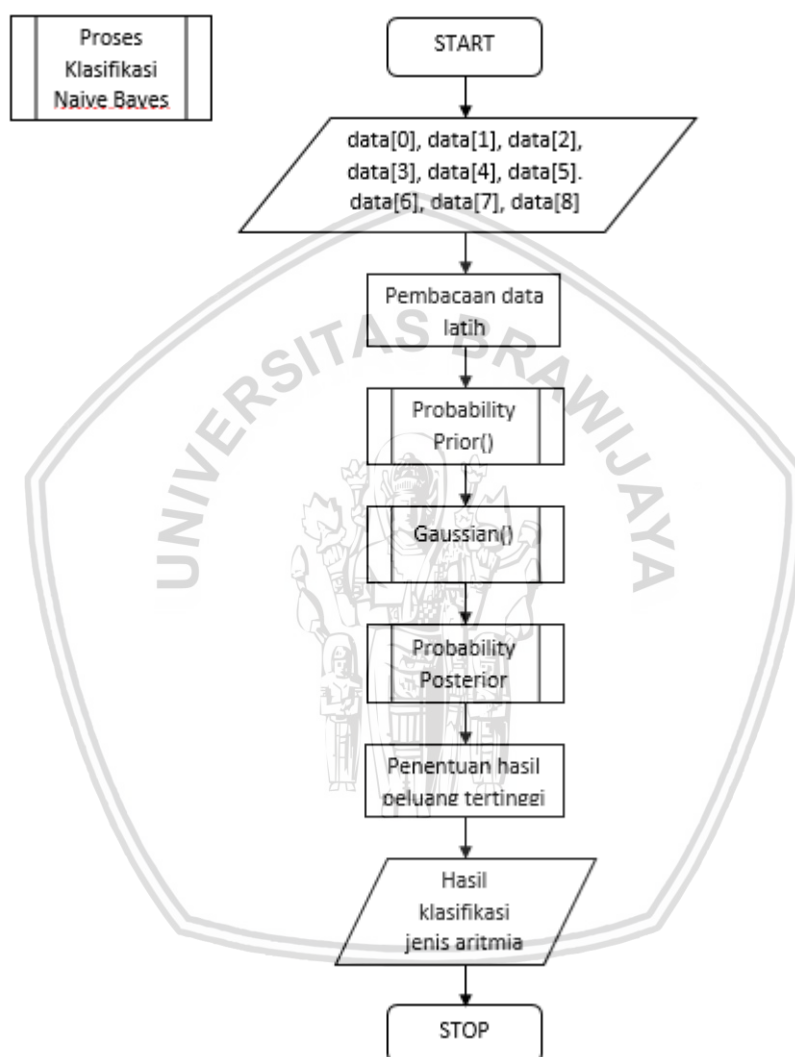
mean : nilai rata-rata dari nilai BPM yang didapatkan untuk dijadikan fitur klasifikasi *Naïve Bayes*.

meanbpm : nilai BPM dari beat ke-1 hingga beat ke-8

Setelah nilai mean didapatkan maka sistem akan mendefinisikan nilai data yang digunakan untuk fitur klasifikasi *Naïve Bayes* dimana nilai dari data[0] sampai dengan data[7] adalah nilai dari R interval dan nilai dari data[8] adalah nilai dari rata-rata BPM yang didapatkan dari **Persamaan (5.2)**. Kemudian masuk ke dalam kondisi jika sistem tidak dimatikan maka sistem akan kembali membaca sensor, jika dimatikan maka sistem akan berhenti.

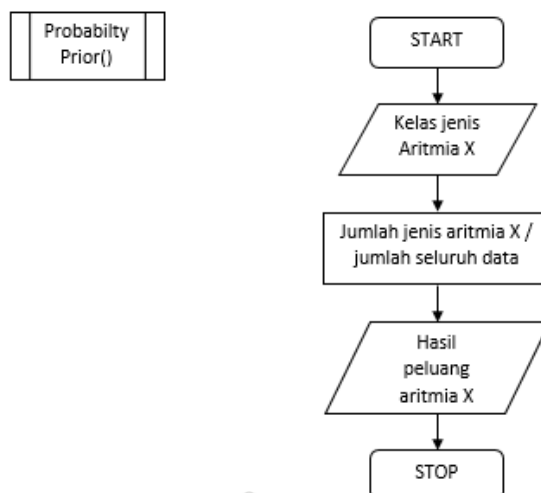
5.1.3.3 Perancangan Klasifikasi *Naïve Bayes*

Setelah nilai dari R interval dan BPM yang merupakan fitur dari klasifikasi *Naïve Bayes* telah didapatkan, maka tahap berikutnya adalah melakukan perancangan klasifikasi *Naïve Bayes*. Klasifikasi *Naïve Bayes* pada sistem ini memiliki 46 data latih didapatkan dari MIT-BIH *Arrhythmia Database* (Goldberger AL, 2000) dan 14 data uji. Klasifikasi *Naïve Bayes* dapat digambarkan menggunakan diagram alir yang dapat dilihat pada Gambar 5.6.



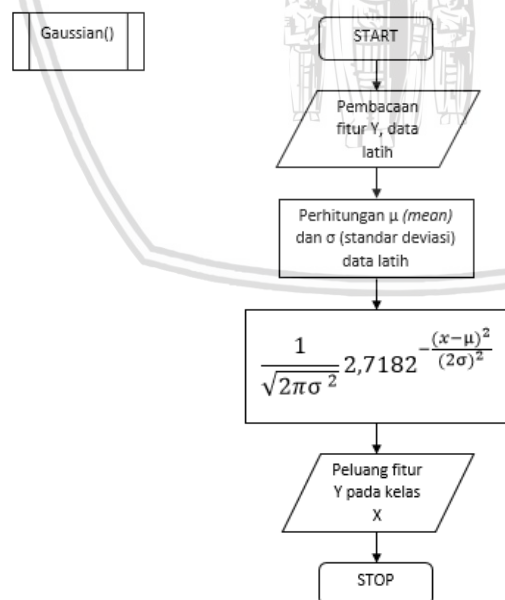
Gambar 5.6 Diagram Alir Perancangan Klasifikasi *Naïve Bayes*

Pada sistem ini, untuk menentukan hasil klasifikasi jenis aritmia menggunakan metode *Naïve Bayes* terdapat beberapa tahap yang perlu dijalankan sesuai dengan diagram alir pada Gambar 5.6 dimana masukan atau fitur yang digunakan adalah nilai R Interval dan nilai BPM yang telah didapat sebelumnya. Selain itu, hasil klasifikasi pada sistem ini juga dipengaruhi oleh nilai data latih yang telah diinisialisasikan sebelumnya. Setelah membaca data latih, sistem akan menjalankan fungsi `ProbabilityPrior()` yang dapat dijabarkan pada Gambar 5.7.



Gambar 5.7 Diagram Alir fungsi ProbabilityPrior()

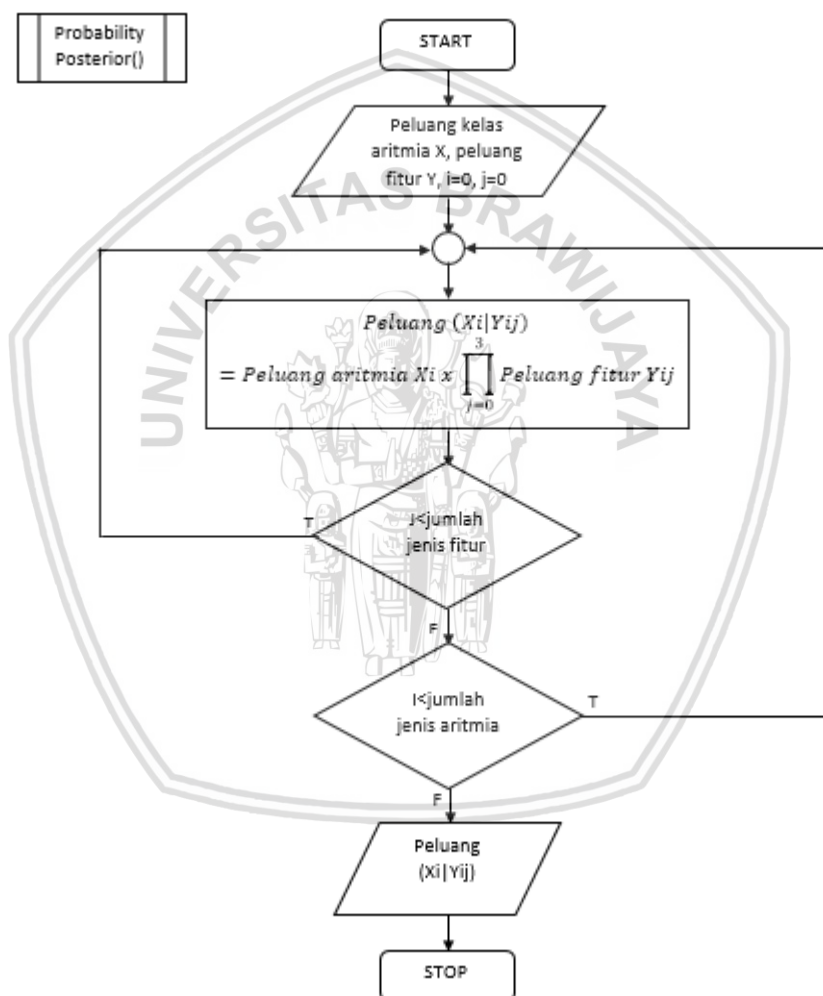
Fungsi ProbabilityPrior() ini merupakan fungsi yang digunakan untuk menghitung peluang pada masing-masing kelas jenis aritmia. Perhitungan peluang pada masing-masing kelas jenis aritmia ini adalah dengan membagi jumlah data dalam suatu kelas (pada sistem ini terdapat tiga kelas, yaitu normal, *bigeminy* dan *trigeminy*) dengan jumlah seluruh data yang ada. Data yang digunakan untuk menghitung nilai *prior* ini merupakan data latih yang telah didapatkan sebelumnya. Tahap berikutnya setelah mendapatkan nilai *prior* dari masing-masing kelas adalah dengan memanggil fungsi Gaussian yang dapat dijabarkan pada Gambar 5.8.



Gambar 5.8 Diagram Alir Fungsi Gaussian()

Fungsi dari gaussian() ini digunakan untuk menentukan nilai peluang dari masing-masing fitur pada suatu kelas. Terdapat sembilan fitur yang digunakan

pada sistem ini yaitu delapan fitur dari nilai R Interval dan satu fitur dari nilai BPM. Sebelum menentukan nilai peluang dari masing-masing fitur pada suatu kelas, sistem harus melakukan perhitungan untuk mencari nilai *mean* dan standar deviasi dari data latih dengan menggunakan **Persamaan (2.4)** dan **Persamaan (2.5)**. Pada sistem ini, data latih berupa nilai *mean* dan standar deviasi disimpan pada mikrokontroler Arduino Uno untuk mempermudah sistem mengakses nilai tersebut pada saat sistem dijalankan. Kemudian sistem akan melakukan perhitungan *Gaussian* dengan menggunakan **Persamaan (2.3)** dimana nilai Y merupakan nilai fitur dari R interval dan BPM. Setelah nilai peluang tiap fitur telah didapatkan, maka selanjutnya sistem akan menjalankan fungsi ProbabilityPosterior() yang dapat digambarkan pada Gambar 5.9.



Gambar 5.9 Diagram Alir Fungsi ProbabilityPosterior()

Fungsi ProbabilityPosterior() merupakan fungsi untuk menentukan nilai dari peluang *posterior*. Peluang *posterior* adalah peluang yang menentukan besarnya peluang dari masing-masing kelas ketika terdapat masukan dari tiap fitur. Pada sistem ini peluang *posterior* digunakan untuk menentukan besar peluang masing-masing jenis PVC Aritmia ketika nilai R interval dan BPM telah

berhasil didapatkan. Proses dari peluang *posterior* adalah dengan melakukan perkalian antara hasil dari fungsi *ProbabilityPrior()* dengan fungsi *Gaussian()*.

Setelah nilai dari peluang *posterior* pada masing-masing kelas telah didapatkan, sistem akan membandingkan nilai peluang *posterior* tiap kelas tersebut. Kelas jenis aritmia dengan nilai tertinggi merupakan hasil akhir dari klasifikasi *Premature Ventricular Contraction Aritmia* yang dideteksi oleh sistem.

Perhitungan untuk mencari nilai *prior*, *mean* dan *standar deviasi* seluruhnya pada sistem ini adalah sebagai berikut.

1. Menghitung nilai peluang *prior* pada masing-masing jenis (kelas) *Premature Ventricular Contraction Aritmia*

$$P_{normal} = \frac{\text{jumlah jenis normal}}{\text{jumlah seluruh data}} = \frac{11}{46} = 0,239130435$$

$$P_{bigeminy} = \frac{\text{jumlah jenis bigeminy}}{\text{jumlah seluruh data}} = \frac{28}{46} = 0,608695652$$

$$P_{trigeminy} = \frac{\text{jumlah jenis trigeminy}}{\text{jumlah seluruh data}} = \frac{7}{46} = 0,152173913$$

2. Menghitung nilai *mean* dan *standar deviasi* dari masing-masing jenis (kelas) *Premature Ventricular Contraction Aritmia*. Sebagai contoh perhitungan dibawah merupakan perhitungan tiap fitur pada kelas *trigeminy*.

a. *Mean* :

$$\begin{aligned}\bar{x}_{RRt7(trigeminy)} &= \frac{\sum_{i=1}^n RRt7_i}{\text{jumlah jenis aritmia ventrikular trigeminy}} \\ &= \frac{519 + 553 + 544 + 536 + 783 + 517 + 486}{7} \\ &= 562,5714286\end{aligned}$$

$$\begin{aligned}\bar{x}_{RRt6(trigeminy)} &= \frac{\sum_{i=1}^n RRt6_i}{\text{jumlah jenis aritmia ventrikular trigeminy}} \\ &= \frac{1331 + 1297 + 1406 + 1356 + 1378 + 997 + 1294}{7} \\ &= 1294,142857\end{aligned}$$

$$\begin{aligned}\bar{x}_{RRt5(trigeminy)} &= \frac{\sum_{i=1}^n RRt5_i}{\text{jumlah jenis aritmia ventrikular trigeminy}} \\ &= \frac{886 + 869 + 903 + 936 + 1036 + 725 + 822}{7} \\ &= 882,4285714\end{aligned}$$

$$\begin{aligned}\bar{x}_{RRt4(trigeminy)} &= \frac{\sum_{i=1}^n RRt4_i}{\text{jumlah jenis aritmia ventrikular trigeminy}} \\ &= \frac{514 + 508 + 547 + 528 + 644 + 486 + 475}{7} \\ &= 528,8571429\end{aligned}$$

$$\begin{aligned}\bar{x}_{RRt3(trigeminy)} &= \frac{\sum_{i=1}^n RRt3_i}{\text{jumlah jenis aritmia ventrikular trigeminy}} \\ &= \frac{1300 + 1264 + 1381 + 1322 + 1433 + 1039 + 864}{7} \\ &= 1229\end{aligned}$$

$$\begin{aligned}\bar{x}_{RRt2(trigeminy)} &= \frac{\sum_{i=1}^n RRt2_i}{\text{jumlah jenis aritmia ventrikular trigeminy}} \\ &= \frac{886 + 858 + 986 + 928 + 1042 + 783 + 747}{7} \\ &= 890\end{aligned}$$

$$\begin{aligned}\bar{x}_{RRt1(trigeminy)} &= \frac{\sum_{i=1}^n RRt1_i}{\text{jumlah jenis aritmia ventrikular trigeminy}} \\ &= \frac{536 + 911 + 536 + 544 + 825 + 536 + 458}{7} \\ &= 620,8571429\end{aligned}$$

$$\begin{aligned}\bar{x}_{RRt(trigeminy)} &= \frac{\sum_{i=1}^n RRt_i}{\text{jumlah jenis aritmia ventrikular trigeminy}} \\ &= \frac{1264 + 911 + 1356 + 1333 + 1364 + 1000 + 886}{7} \\ &= 1159,142857\end{aligned}$$

$$\begin{aligned}\bar{x}_{BPM(trigeminy)} &= \frac{\sum_{i=1}^n BPM_i}{\text{jumlah jenis aritmia ventrikular trigeminy}} \\ &= \frac{66 + 66 + 62 + 64 + 56 + 78 + 79}{7} \\ &= 67,28571429\end{aligned}$$

Perhitungan yang sama dilakukan pada kelas jenis normal dan *bigeminy* untuk mencari nilai *mean* pada masing-masing fitur pada kelas tersebut. Hasil

dari perhitungan *mean* seluruhnya pada sistem ini dapat ditunjukkan pada Tabel 5.4.

Tabel 5.4 Nilai *Mean* Pada Setiap Jenis Fitur dan Kelas Aritmia

	Normal	Bigeminy	Trigeminy
RR_{t-7}	787,6363636	522,535714	562,5714286
RR_{t-6}	805,1818182	928,464286	1294,142857
RR_{t-5}	791,5454545	519,464286	882,4285714
RR_{t-4}	799	945,5	528,8571429
RR_{t-3}	807,3636364	526,75	1229
RR_{t-2}	803,2727273	917,464286	890
RR_{t-1}	779,4545455	508,535714	620,8571429
RR_t	803,5454545	881,214286	1159,142857
BPM	76,18181818	83,4642857	67,28571429

b. Standar Deviasi :

$$\sigma_{RRt-7(trigeminy)} = \frac{\sqrt{\sum_{i=1}^n (RRt - 7_i - \bar{x}_{RRt-7(trigeminy)})^2}}{\sqrt{\text{Jumlah kelas aritmia trigeminy} - 1}}$$

$$= \frac{\sqrt{(519-562,5714286)^2 + (553-562,5714286)^2 + (544-562,5714286)^2 + (536-562,5714286)^2 + (783-562,5714286)^2 + (517-562,5714286)^2 + (486-562,5714286)^2}}{\sqrt{7-1}}$$

$$= 99,64078$$

$$\sigma_{RRt-6(trigeminy)} = \frac{\sqrt{\sum_{i=1}^n (RRt - 6_i - \bar{x}_{RRt-6(trigeminy)})^2}}{\sqrt{\text{Jumlah kelas aritmia trigeminy} - 1}}$$

$$= \frac{\sqrt{(1331-1294,142857)^2 + (1297-1294,142857)^2 + (1406-1294,142857)^2 + (1356-1294,142857)^2 + (1378-1294,142857)^2 + (997-1294,142857)^2 + (1294-1294,142857)^2}}{\sqrt{7-1}}$$

$$= 137,2533$$

$$\sigma_{RRt-5(trigeminy)} = \frac{\sqrt{\sum_{i=1}^n (RRt - 5_i - \bar{x}_{RRt-5(trigeminy)})^2}}{\sqrt{\text{Jumlah kelas aritmia trigeminy} - 1}}$$

$$= \sqrt{\frac{(886-882,4285714)^2 + (869-882,4285714)^2 + (903-882,4285714)^2 + (936-882,4285714)^2 + (1036-882,4285714)^2 + (725-882,4285714)^2 + (822-882,4285714)^2}{7-1}}$$

$$= 96,18187$$

$$\sigma_{RRt-4(trigeminy)} = \sqrt{\frac{\sum_{i=1}^n (RRt - 4_i - \bar{x}_{RRt-4(trigeminy)})^2}{Jumlah\ kelas\ aritmia\ trigeminy - 1}}$$

$$= \sqrt{\frac{(514-528,8571429)^2 + (508-528,8571429)^2 + (547-528,8571429)^2 + (528-528,8571429)^2 + (644-528,8571429)^2 + (486-528,8571429)^2 + (475-528,8571429)^2}{7-1}}$$

$$= 56,24479$$

$$\sigma_{RRt-3(trigeminy)} = \sqrt{\frac{\sum_{i=1}^n (RRt - 3_i - \bar{x}_{RRt-3(trigeminy)})^2}{Jumlah\ kelas\ aritmia\ trigeminy - 1}}$$

$$= \sqrt{\frac{(1300-1229)^2 + (1264-1229)^2 + (1381-1229)^2 + (1322-1229)^2 + (1433-1229)^2 + (1039-1229)^2 + (864-1229)^2}{7-1}}$$

$$= 203,6991$$

$$\sigma_{RRt-2(trigeminy)} = \sqrt{\frac{\sum_{i=1}^n (RRt - 2_i - \bar{x}_{RRt-2(trigeminy)})^2}{Jumlah\ kelas\ aritmia\ trigeminy - 1}}$$

$$= \sqrt{\frac{(886-890)^2 + (858-890)^2 + (986-890)^2 + (928-890)^2 + (1042-890)^2 + (783-890)^2 + (747-890)^2}{7-1}}$$

$$= 105,4372$$

$$\sigma_{RRt-1(trigeminy)} = \sqrt{\frac{\sum_{i=1}^n (RRt - 1_i - \bar{x}_{RRt-1(trigeminy)})^2}{Jumlah\ kelas\ aritmia\ trigeminy - 1}}$$

$$= \sqrt{\frac{(536-620,8571429)^2 + (911-620,8571429)^2 + (536-620,8571429)^2 + (544-620,8571429)^2 + (825-620,8571429)^2 + (536-620,8571429)^2 + (458-620,8571429)^2}{7-1}}$$

$$= 173,1516$$

$$\sigma_{RRt(trigeminy)} = \sqrt{\frac{\sum_{i=1}^n (RRt_i - \bar{x}_{RRt(trigeminy)})^2}{Jumlah\ kelas\ aritmia\ trigeminy - 1}}$$

$$\begin{aligned}
 &= \sqrt{\frac{(1264-1159,142857)^2 + (911-1159,142857)^2 + (1356-1159,142857)^2 + (1333-1159,142857)^2 + (1364-1159,142857)^2 + (1000-1159,142857)^2 + (886-1159,142857)^2}{7-1}} \\
 &= 217,351 \\
 \sigma_{BPM(trigeminy)} &= \sqrt{\frac{\sum_{i=1}^n (RRt_i - \bar{x}_{BPM(trigeminy)})^2}{Jumlah\ kelas\ aritmia\ trigeminy - 1}} \\
 &= \sqrt{\frac{(66-67,28571429)^2 + (66-67,28571429)^2 + (62-67,28571429)^2 + (64-67,28571429)^2 + (56-67,28571429)^2 + (78-67,28571429)^2 + (79-67,28571429)^2}{7-1}} \\
 &= 8,380817
 \end{aligned}$$

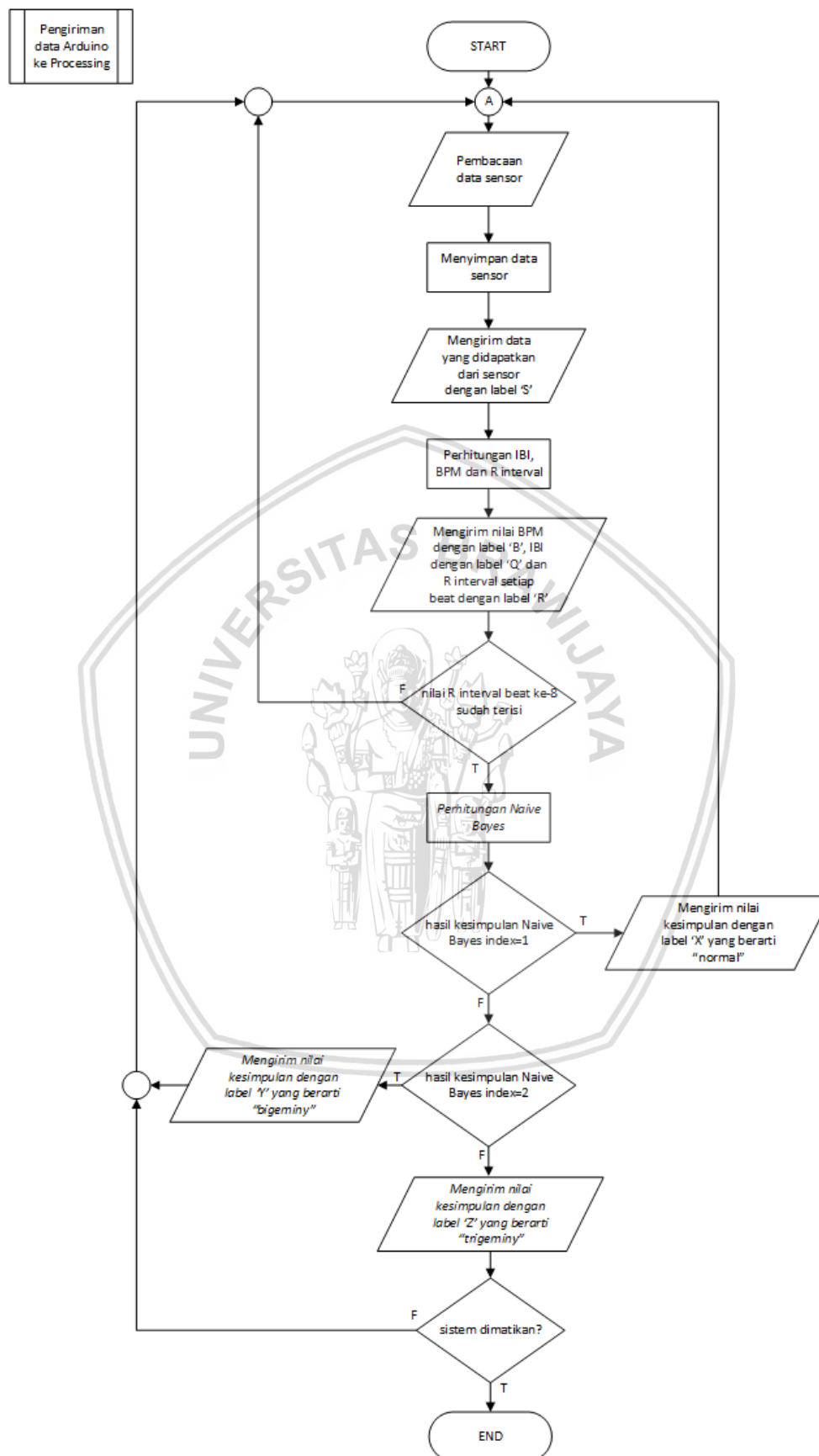
Perhitungan yang sama dilakukan untuk mencari standar deviasi setiap fitur jenis normal dan jenis *Premature Ventricular Contraction* Aritmia *bigeminy*. Hasil seluruh perhitungan standar deviasi jenis *Premature Ventricular Contraction* Aritmia dapat dilihat pada Tabel 5.5.

Tabel 5.5 Nilai Standar Deviasi Pada Setiap Jenis Fitur dan Kelas Aritmia

	Normal	Bigeminy	Trigeminy
RR _{t-7}	94,28629	79,15736	99,64078
RR _{t-6}	104,0835	133,4234	137,2533
RR _{t-5}	110,9726	64,54373	96,18187
RR _{t-4}	116,7484	123,8541	56,24479
RR _{t-3}	119,3759	64,79008	203,6991
RR _{t-2}	116,3594	145,3474	105,4372
RR _{t-1}	120,9776	69,59314	173,1516
RR _t	121,0538	145,4103	217,351
BPM	10,78157	6,379612	8,380817

5.1.3.4 Perancangan Pengiriman Data Arduino Ke Processing

Perancangan pengiriman data dari Arduino ke Processing ini menjelaskan proses perancangan yang dilakukan untuk mengirimkan data dari Arduino ke Processing. Perancangan pengiriman data dari Arduino ke Processing ini dapat dijabarkan menggunakan diagram alir pada Gambar 5.10.

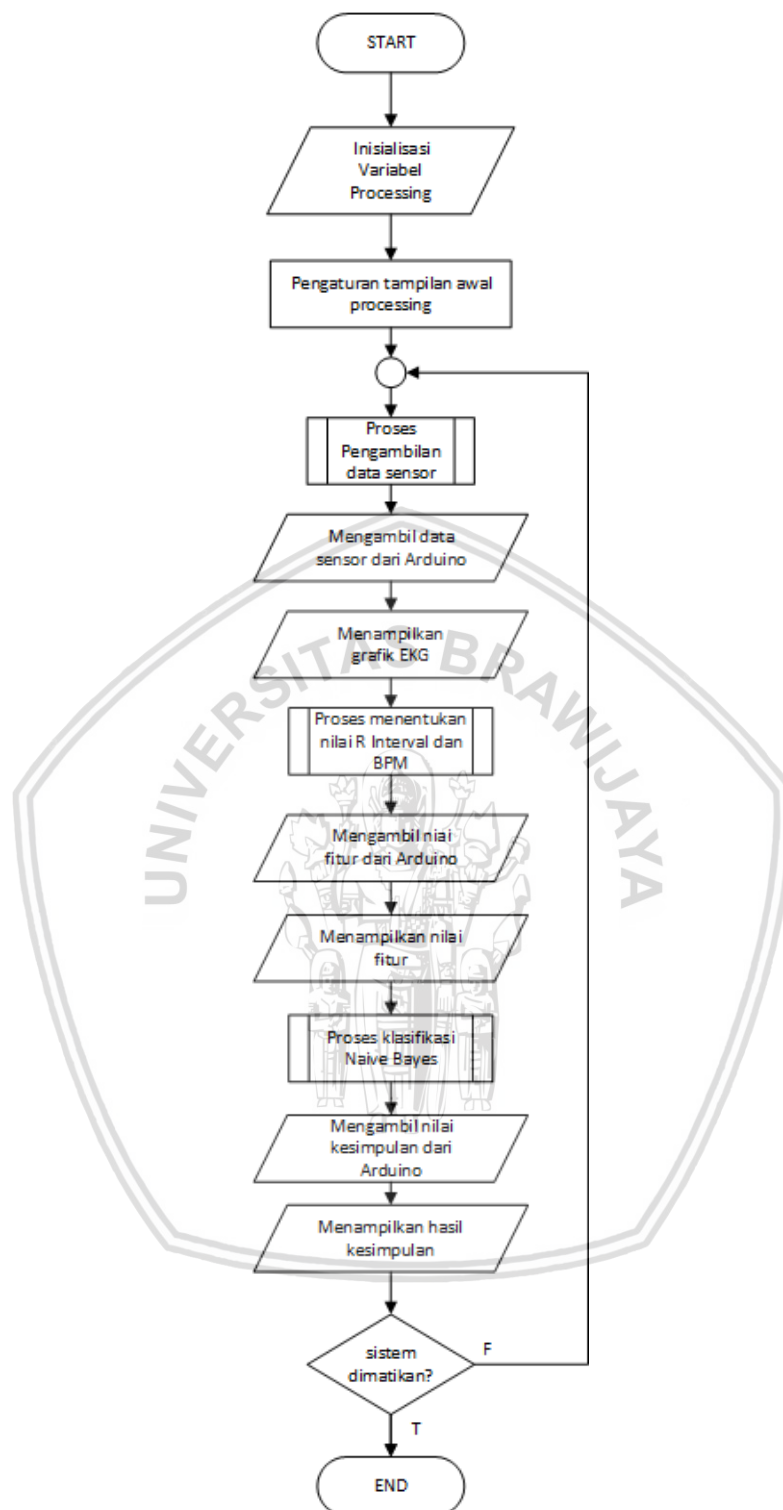


Gambar 5.10 Diagram Alir Pengiriman Data Arduino ke Processing

Untuk dapat mengirimkan data dari Arduino ke Processing, sistem terlebih dahulu melakukan pembacaan data pada sensor dan menyimpan data tersebut. Kemudian, sistem mengirimkan data sensor ke Processing dengan label "S". Selanjutnya sistem akan melakukan perhitungan BPM, IBI seperti yang dijelaskan pada diagram alir Gambar 5.5 sebelumnya. Setelah nilai BPM dan IBI didapatkan, sistem akan mengirimkan data BPM dengan label "B" dan IBI dengan label "Q" ke Processing. Masuk ke dalam kondisi, jika nilai R interval tidak terisi sebanyak 8 kali, maka sistem akan kembali membaca nilai data pada sensor. Jika nilai R interval terisi sebanyak 8 kali maka sistem akan masuk ke dalam perhitungan *Naïve Bayes*. Perhitungan *Naïve Bayes* yang dilakukan sesuai seperti yang dijelaskan pada diagram alir Gambar 5.6 di atas. Penentuan hasil kesimpulan dari klasifikasi metode *Naïve Bayes* pada sistem dilakukan perbandingan setiap kelas dimana jenis normal merupakan index dengan nilai 1, bigeminy merupakan index dengan nilai 2 dan trigeminy merupakan index dengan nilai 3. Jika nilai index yang didapatkan pada penentuan hasil kesimpulan bernilai 1 maka Arduino akan mengirimkan hasil kesimpulan dengan label "X" yang berarti normal, jika nilai index bernilai 2 maka Arduino akan mengirimkan hasil kesimpulan dengan label "Y" yang berarti bigeminy dan jika nilai index bernilai 3 maka Arduino akan mengirimkan hasil kesimpulan dengan label "Z" yang berarti trigeminy. Setelah hasil kesimpulan didapatkan sistem akan kembali membaca nilai data pada sensor.

5.1.3.5 Perancangan Menampilkan Hasil Keluaran Menggunakan Processing

Perancangan menampilkan hasil keluaran menggunakan Processing ini menjelaskan mengenai rancangan yang dilakukan agar sistem dapat menampilkan hasil keluaran yang diinginkan dengan menggunakan Processing. Perancangan untuk menampilkan hasil keluaran dari sistem dengan menggunakan Processing dapat digambarkan menggunakan diagram alir pada Gambar 5.11.



Gambar 5.11 Diagram Alir Menampilkan Hasil Keluaran Menggunakan Processing

Untuk menampilkan hasil keluaran menggunakan perangkat lunak Processing ini, terdapat beberapa tahap yang diperlukan seperti yang dapat dilihat pada Gambar 5.11. Tahap pertama yang dilakukan adalah melakukan inisialisasi variabel Processing yang berfungsi untuk melakukan inisialisasi dan deklarasi

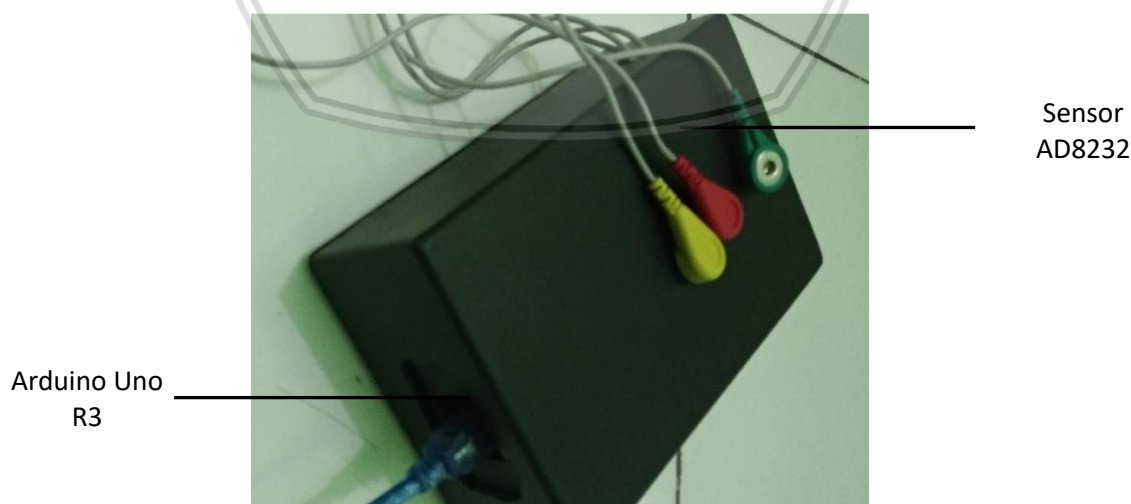
variabel yang akan digunakan untuk menampilkan hasil keluaran pada sistem. Tahap selanjutnya yaitu melakukan pengaturan tampilan awal atau *interface* yang digunakan oleh sistem. Tahap ketiga merupakan proses pengambilan data sensor yang telah dilakukan sebelumnya. Setelah data sensor didapatkan, Arduino akan mengirim data tersebut ke Processing dan Processing akan menampilkan grafik EKG dari nilai sensor tersebut. Kemudian sistem melakukan proses mencari nilai R Interval dan BPM. Begitu nilai fitur tersebut didapatkan, Arduino akan mengirim data tersebut ke Processing dan Processing akan menampilkan nilai fitur yang telah didapat tersebut. Terakhir sistem akan melakukan proses klasifikasi *Naive Bayes*. Begitu hasil klasifikasi didapatkan, Arduino akan mengirim hasil klasifikasi tersebut ke Processing, kemudian Processing akan menampilkan hasil dari klasifikasi *Naive Bayes* tersebut. Setelah hasil kesimpulan berhasil ditampilkan, jika sistem tidak dimatikan maka sistem akan kembali ke proses membaca data sensor.

5.2 Implementasi Sistem

Pada Sub-bab ini akan dijelaskan mengenai tahap implementasi sistem untuk merealisasikan pembuatan sistem dengan berdasarkan seluruh perancangan yang telah dilakukan sebelumnya. Pada implementasi sistem ini menjelaskan secara rinci mengenai implementasi prototype, implementasi perangkat keras dan implementasi perangkat lunak dari sistem.

5.2.1 Implementasi *Prototype* Sistem

Pada tahap ini akan dijelaskan mengenai proses implementasi desain pada alat pendeteksi *Premature Ventricular Contraction Aritmia* dengan mengacu pada perancangan yang telah dijelaskan pada perancangan implementasi desain alat pendeteksi *Premature Ventricular Contraction Aritmia* di atas. Hasil implementasi desain alat pendeteksi *Premature Ventricular Contraction Aritmia* ini dapat dilihat pada Gambar 5.12.



Gambar 5.12 Implementasi *Prototype* Alat Pendeteksi *Premature Ventricular Contraction Aritmia*

Implementasi prototype alat pendeteksi *Premature Ventricular Contraction Aritmia* ini menggunakan kotak hitam dengan ukuran 15x10x5 cm³. Kotak hitam ini memiliki bahan akrilik dengan ketebalan 2 mm.

5.2.2 Implementasi Perangkat Keras Sistem

Tahap ini menjelaskan mengenai proses implementasi perangkat keras dari sistem yang mencakup komponen-komponen elektronik sistem.

5.2.2.1 Implementasi Mikrokontroler dan Sensor

Tahap ini menjelaskan mengenai proses implementasi dari mikrokontroler Arduino Uno dan sensor AD8232 Heart Rate dengan hasil keluaran yang ditampilkan pada laptop. Hasil implementasi perangkat keras alat pendeteksi *Premature Ventricular Contraction Aritmia* ditunjukkan pada Gambar 5.13.

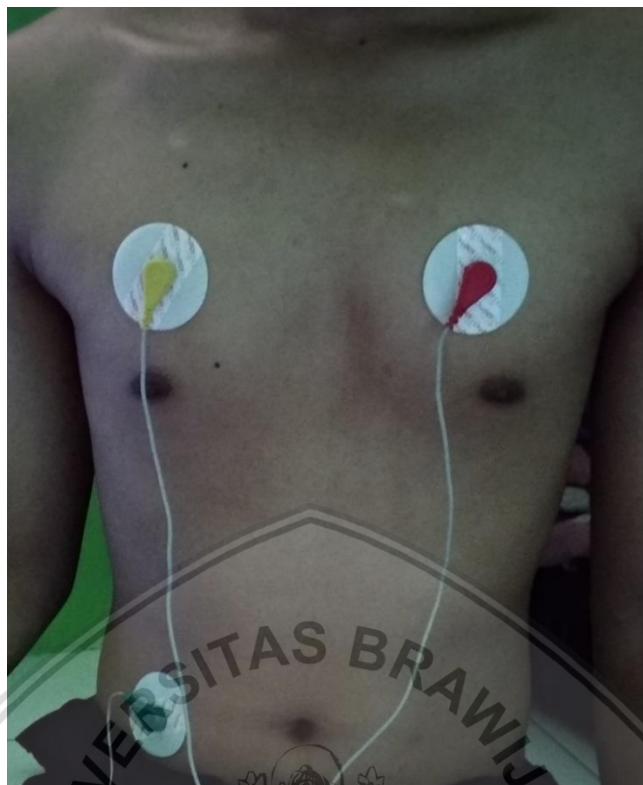


Gambar 5.13 Implementasi Perangkat Keras Sistem

Gambar 5.13 menunjukkan bahwa Arduino Uno dan sensor AD8232 dihubungkan menggunakan kabel *jumper* sesuai dengan perancangan yang dilakukan sebelumnya.

5.2.2.2 Implementasi Mikrokontroler dan Sensor

Pada tahap ini menjelaskan mengenai proses implementasi dari penempatan elektroda dari Sistem Pendeteksi *Premature Ventricular Contraction Aritmia* dengan berdasarkan teori 3 Lead. Implementasi dari penempatan elektroda dari Sistem Pendeteksi *Premature Ventricular Contraction Aritmia* ini dapat dilihat pada Gambar 5.14.



Gambar 5.14 Implementasi Penempatan Elektroda

Gambar 5.14 elektroda dengan warna kuning ditempatkan pada tangan kanan, elektroda dengan warna merah ditempatkan pada tangan kiri dan elektroda dengan warna hijau ditempatkan pada perut bagian kanan sesuai dengan yang dijelaskan pada perancangan penempatan elektroda di atas.

5.2.3 Implementasi Perangkat Lunak Sistem

Implementasi perangkat lunak alat pendeteksi *Premature Ventricular Contraction Aritmia* ini menjelaskan mengenai proses realisasi program yang digunakan dengan berdasarkan pada perancangan yang telah dibuat pada sub-bab 5.1.3. Implementasi perangkat lunak pada sistem ini memiliki dua kode program dimana kode program pada Arduino IDE 1.8.5 digunakan untuk tahap pemrosesan pada sistem, dan kode program pada Processing 2.2.1 yang digunakan untuk menampilkan hasil keluaran pada sistem. Kode program pada Arduino dilakukan inisialisasi library yang memiliki fungsi tertentu. Pengimplementasian library pada sistem ini ditunjukkan pada Tabel 5.6.

Tabel 5.6 Kode Program Inisialisasi *library* Sistem Pendeteksi *Premature Ventricular Contraction Aritmia*

Baris	Kode Program
1	<code>#include <FIR.h></code>
2	<code>#include <math.h></code>

Pada Tabel 5.6 dapat dilihat library yang digunakan pada sistem ini adalah library “FIR.h” dengan fungsi filter pada sensor AD8232 dan library “math.h” dengan fungsi melakukan perhitungan matematika kompleks.

5.2.3.1 Implementasi Kode Program Pengambilan Data Sensor

Dalam pengimplementasian untuk membaca nilai dari sensor AD8232 pada sistem ini terlebih dahulu dilakukan inisialisasi variabel dan konfigurasi pin sensor berdasarkan perancangan perangkat keras dan perancangan perangkat lunak sistem yang telah dijelaskan sebelumnya. Pengimplementasian kode program yang digunakan untuk mengambil data sensor pada sistem ini dapat dilihat pada Tabel 5.7.

Tabel 5.7 Kode Program Inisialisasi Variabel Pembacaan Sensor

Baris	Kode Program
1	#define PROCESSING_VISUALIZER 1
2	#define SERIAL_PLOTTER 2
3	int pulsePin = 0;
4	volatile float BPM; // menyimpan nilai bpm
5	volatile float Signal; // masukan data dari
6	sensor
7	volatile float RR[8]; // untuk mencari nilai
8	RRt-7 sampai RRt
9	volatile float IBI = 600.0; // menyimpan waktu
10	interval tiap beat
11	float data[9];
12	volatile float meanbpm[8];
13	volatile float mean;
14	volatile boolean Pulse = false; // True ketika detak jantung
15	pengguna terdeteksi secara live
16	volatile boolean QS = false; // True ketika arduino
17	menemukan beat
18	byte indexRR = 0;
19	byte indexbpm = 0;
20	
21	volatile float rate[10];
22	volatile unsigned long sampleCounter = 0;
23	volatile unsigned long lastBeatTime = 0;
24	volatile float P = 512;
25	volatile float T = 512;
26	volatile float thresh = 530;
27	volatile float amp = 0;
28	volatile boolean firstBeat = true;
29	volatile boolean secondBeat = false;
30	

Baris	Kode Program
1	<code>void interruptSetup(){</code>
2	<code>...</code>
3	<code>// Interrupt Service Routine Timer 2 memastikan sistem membaca</code>
4	<code>setiap 2ms</code>
5	<code>ISR(TIMER2_COMPA_vect){</code>
6	<code>cli();</code> <code>// mematikan</code>
7	<code>interrupt</code>

8	Signal = analogRead(pulsePin);	// membaca ad8232
9	sampleCounter += 2;	// menyimpan nilai
10	waktu dengan satuan mS	
11	int N = sampleCounter - lastBeatTime;	// monitoring
12	waktu dari beat terakhir untuk menghindari noise	
13	...	

Kode program pembacaan nilai sensor ad8232 pada sistem ini terdapat pada kelas interruptSetup(). Seperti yang dijelaskan sebelumnya, sistem menggunakan Timer2 yang digunakan agar sistem dapat membaca sensor setiap 2 milisekon. Pada baris ke-6 fungsi dari cli() adalah untuk menyalakan atau mematikan global interrupt yang digunakan pada sistem. Kemudian baris ke-8 hingga baris ke-11 digunakan untuk pembacaan data sensor pada sistem dengan menyimpan waktu setiap sistem membaca data pada sensor.

5.2.3.2 Implementasi Kode Program Penentuan Nilai R Interval dan BPM

Setelah sistem mendapatkan data pada sensor AD8232, sistem akan melakukan proses mencari titik puncak dan titik terendah (QRS Complex) dan menghitung waktu antara setiap titik puncak tersebut. Pengimplementasian kode program yang digunakan untuk mencari nilai titik tertinggi dan titik terendah gelombang pada sistem ini dapat dilihat pada Tabel 5.9.

Tabel 5.9 Kode Program Mencari Nilai Titik Terendah dan Titik Tertinggi Gelombang

Baris	Kode Program
1	ISR(TIMER2_COMPA_vect){
2	...
3	// mencari titik puncak
4	if(Signal < thresh && N > (IBI/5)*3){ // menghindari
5	dichrotic noise dengan menunggu 3/5 dari IBI
6	if (Signal < T){
7	T = Signal; // mencari titik
8	terendah pada gelombang
9	}
10	}
11	
12	if(Signal > thresh && Signal > P){ // mencari titik
13	tertinggi pada gelombang
14	P = Signal; // P merupakan titik
15	puncak
16	}
17	...

Kode program untuk mencari titik tertinggi dan titik terendah pada sistem ini berada pada fungsi ISR atau yang biasa disebut dengan Interrupt Service Routine yang merupakan fungsi untuk melakukan interrupt ketika suatu kondisi telah terpenuhi. Pada baris ke-5 merupakan kondisi yang digunakan untuk menghindari dichrotic noise dengan menunggu waktu 3/5 dari IBI atau Inter-Beat Interval. Kemudian pada baris ke-7 hingga ke-10 merupakan kondisi untuk mencari titik terendah pada gelombang dan baris ke-13 hingga baris ke-15 merupakan kondisi untuk mencari titik tertinggi pada gelombang.

Ketika titik tertinggi berhasil didapatkan, maka tahap selanjutnya adalah mencari nilai interval dari R atau nilai interval dari titik puncak yang didapatkan sebelumnya. Pengimplementasian kode program yang digunakan untuk mencari nilai R interval pada sistem ini dapat dilihat pada Tabel 5.10.

Tabel 5.10 Kode Program Mencari Nilai R Interval dan BPM

Baris	Kode Program
1	ISR(TIMER2_COMPA_vect){
2	...
3	if (N > 250){ // menghindari
4	noise apabila frekuensi terlalu tinggi
5	if ((Signal > thresh) && (Pulse == false) && (N >
6	(IBI/5)*3)){
7	Pulse = true; // set Pulse
8	flag ketika terdapat detak jantung
9	IBI = sampleCounter - lastBeatTime; // menghitung
10	waktu setiap beat dengan satuan ms
11	lastBeatTime = sampleCounter; // mengikuti
12	nilai waktu untuk detak selanjutnya
13	BPM = 60000/IBI; // mencari nilai bpm
14	QS = true; // set QS flag
15	...

Kode program untuk mencari nilai R interval dan BPM ini masih berada di dalam fungsi yang sama dengan kode program untuk mencari titik tertinggi dan titik terendah, yaitu fungsi ISR. Baris ke-3 hingga baris ke-11 pada Tabel 5.10 di atas merupakan kondisi yang memiliki fungsi untuk menghitung nilai IBI dengan satuan milisekon dan mendefinisikan nilai lastBeatTime menjadi nilai sampleCounter. Kemudian baris ke-13 digunakan untuk menghitung nilai BPM dengan menggunakan **Persamaan (5.1)**. Contoh dari perhitungan nilai BPM dengan nilai IBI=957 ms adalah sebagai berikut.

$$[BPM] = \frac{60000}{IBI}$$

$$[BPM] = \frac{60000}{957}$$

[BPM] = 63

Setelah nilai dari IBI atau R interval dan BPM didapatkan, sistem akan menyimpan nilai tersebut dan dijadikan sebagai fitur untuk metode klasifikasi *Naïve Bayes*. Pengimplementasian kode program untuk menentukan nilai R interval dan BPM yang dijadikan sebagai fitur untuk metode klasifikasi *Naïve Bayes* pada sistem ini dapat dilihat pada Tabel 5.11.

Tabel 5.11 Kode Program Penentuan Nilai R Interval dan BPM Sebagai Fitur Klasifikasi *Naïve Bayes*

Baris	Kode Program
1	ISR(TIMER2_COMPA_vect) {
2	...
3	RR[indexRR] = IBI; //menyimpan nilai IBI
4	ke RR
5	meanbpm[indexbpm] = BPM; //menyimpan nilai BPM
6	ke meanbpm
7	indexRR++;
8	indexbpm++;
9	
10	if(indexRR == 7) {
11	mean = (meanbpm[0] + meanbpm[1] + meanbpm[2] +
12	meanbpm[3] + meanbpm[4] + meanbpm[5] + meanbpm[6] + meanbpm[7])
13	/ 8; //mencari nilai rata-rata bpm yang digunakan untuk bayes
14	data[0] = RR[0];
15	data[1] = RR[1];
16	data[2] = RR[2];
17	data[3] = RR[3];
18	data[4] = RR[4];
19	data[5] = RR[5];
20	data[6] = RR[6];
21	data[7] = RR[7];
22	data[8] = mean;
23	bayescondition = true;
24	indexRR = 0;
25	indexbpm = 0;
26	}
27	if(bayescondition == true) {
28	bayes(data); //melakukan fungsi
29	bayes apabila flag bayescondition bernilai true
30	bayescondition = false;
31	}
32	}
33	}

```

34
35     if (Signal < thresh && Pulse == true){
36         Pulse = false;                // mereset flag Pulse
37         amp = P - T;                  // mencari nilai
38         amplitudo dari gelombang
39         thresh = amp/2 + T;           // set agar nilai
40         thresh lebih dari 50% nilai amplitudo
41         P = thresh;                  // mereset nilai P
42         dan T
43         T = thresh;
44     }
45
46     if (N > 2500){                    // jika lebih dari
47     2,5 detik tidak ada detak jantung
48         thresh = 530;                // set thresh ke
49         default
50         P = 512;                      // set P ke default
51         T = 512;                      // set T ke default
52         lastBeatTime = sampleCounter;
53         firstBeat = true;
54         secondBeat = false;
55     }
56     sei();                            // enable interrupt
57 } //end isr
58

```

Baris ke-3 hingga baris ke-8 pada Tabel 5.11 di atas berfungsi untuk menyimpan nilai IBI dan BPM yang telah didapatkan sebelumnya ke variabel array RR dan meanbpm. Variabel array RR digunakan untuk menyimpan nilai RR_{t-7} hingga RR_t . Kemudian masuk ke dalam kondisi apabila jantung sudah berdetak selama delapan kali. Pada baris ke-11 hingga baris ke-13 berfungsi untuk mencari nilai rata-rata dari BPM yang dijadikan untuk fitur klasifikasi *Naïve Bayes*. Perhitungan untuk mencari nilai rata-rata dari BPM tersebut dilakukan menggunakan **Persamaan (5.2)** sebelumnya. Contoh dari perhitungan nilai rata-rata BPM atau mean dengan nilai meanbpm1=64, meanbpm2=60, meanbpm3=61, meanbpm4=67, meanbpm5=65, meanbpm6=65, meanbpm7=68, meanbpm8=70 adalah sebagai berikut.

$$[mean] = \frac{\sum_{i=0}^7 meanbpm_i}{8}$$

$$[mean] = \frac{64 + 60 + 61 + 67 + 65 + 65 + 68 + 70}{8}$$

$$[mean] = 65$$

Pada baris ke-14 hingga baris ke-26 berfungsi untuk menyimpan nilai dari variabel array RR dan variabel mean ke dalam variabel array data. Variabel array data ini merupakan masukan yang digunakan untuk klasifikasi metode *Naïve Bayes*. Baris ke-35 hingga baris ke-57 merupakan kondisi yang digunakan ketika jantung berelaksasi.

5.2.3.3 Implementasi Kode Program Klasifikasi *Naïve Bayes*

Tahap ini memiliki tujuan untuk mengimplementasikan kode program untuk melakukan pengambilan keputusan jenis *Premature Ventricular Contraction* Aritmia berdasarkan nilai R Interval dan BPM pada jantung yang dideteksi. Tahap pertama yang dilakukan pada pengimplementasian kode program klasifikasi *Naïve Bayes* pada sistem ini adalah melakukan inisialisasi variabel dan tipe data yang dapat dilihat pada Tabel 5.12.

Tabel 5.12 Kode Program Inisialisasi Variabel *Naïve Bayes*

Baris	Kode Program
1	<code>float gauss[4][9];</code>
2	<code>float hasil[4];</code>
3	<code>float tertinggi = -1.000; //default</code>
4	<code>int index = 0;</code>
5	<code>int gauske = 0;</code>
6	<code>char decision;</code>

Tabel 5.12 menunjukkan inisialisasi dari variabel dan tipe data yang digunakan untuk pemrograman metode klasifikasi *Naïve Bayes* yang digunakan pada Sistem Pendeteksi *Premature Ventricular Contraction* Aritmia ini. Inisialisasi pada variabel tersebut merupakan inisialisasi sebagai variabel global sehingga seluruh fungsi pada Sistem Pendeteksi *Premature Ventricular Contraction* Aritmia ini dapat menggunakan variabel tersebut.

Setelah melakukan inisialisasi, tahap berikutnya adalah menentukan nilai peluang prior pada masing-masing jenis *Premature Ventricular Contraction* Aritmia seperti yang dijelaskan pada diagram alir Gambar 5.5 dan menentukan nilai mean dan standar deviasi pada setiap fitur di setiap kelas seperti yang dijelaskan pada perancangan klasifikasi *Naïve Bayes* sebelumnya. Kode program yang digunakan pada pengimplementasian variabel prior, mean dan standar deviasi dapat dilihat pada Tabel 5.13.

Tabel 5.13 Kode Program Inisialisasi Variabel *Prior*, *Mean* dan Standar Deviasi

Baris	Kode Program
1	<code>float pNormal = 0.239130435; // 11/46</code>
2	<code>float pBigeminy = 0.608695652; // 28/46</code>
3	<code>float pTrigeminy = 0.152173913; // 7/46</code>
4	
5	<code>//mean1-9, standar deviasi 1-9</code>
6	

7	float normal[2][9] = {{787.6363636, 805.1818182, 791.5454545,
8	799, 807.3636364, 803.2727273, 779.4545455, 803.5454545,
9	76.18181818}, {94,28629253, 104.0834733, 110.9725899,
10	116.7484475, 119.3758597, 116.3594424, 120.9775609, 121.053847,
11	10.78156523}};
12	float bigeminy[2][9] = {{522.5357143, 928.4642857, 519.4642857,
13	945.5, 526.75, 917.4642857, 508.5357143, 881.2142857,
14	83.95564441}, {79.15735952, 133.4233862, 64.54372606,
15	123.8540808, 64.79008126, 145.3474344, 69.59313718, 145.4102584,
16	6.363150709}};
17	float trigeminy[2][9] = {{562.5714286, 1294.142857, 882.4285714,
18	528.8571429, 1229, 890, 620.8571429, 1159.142857, 67.85849228},
19	{99.64078339, 137.2533285, 96.18187137, 56.24478812,
20	203.6991245, 105.4371851, 173.1515989, 217.3510437,
	8.501943065}};

Setelah melakukan perhitungan manual untuk menentukan nilai prior, mean dan standar deviasi setiap fitur pada masing-masing kelas, selanjutnya perlu dilakukan inisialisasi untuk menyimpan nilai-nilai tersebut. Dapat dilihat pada Tabel 5.13 di atas terdapat tiga kelas yang digunakan pada sistem ini yaitu normal, bigeminy dan trigeminy dimana nilai prior dari masing-masing kelas tersebut disimpan dalam variabel dengan tipe data float, kemudian nilai mean dan standar deviasi yang disimpan pada variabel array dengan dimensi 2x9 dengan dimensi pertama menyimpan nilai mean setiap fitur dan dimensi kedua menyimpan nilai standar deviasi setiap fitur.

Tahap berikutnya adalah melakukan implementasi untuk melakukan perhitungan gaussian seperti yang dijelaskan pada perancangan diagram alir Gambar 5.6 sebelumnya. Sesuai dengan diagram alir tersebut, implementasi dari kode program perhitungan Gaussian pada sistem ini ditunjukkan pada Tabel 5.14.

Tabel 5.14 Kode Program Fungsi *Gaussian*

Baris	Kode Program
1	void gaussian(float data_uji[9], float data_latih[2][9]) {
2	double a, e, f, gau;
3	for(int i = 0; i < 9; i++) {
4	a = 2 * 3.14 * (pow(data_latih[1][i], 2)); //data_latih[1] =
5	standar deviasi
6	e = -((pow((data_uji[i] - data_latih[0][i]), 2)) / (2 *
7	pow(data_latih[1][i], 2)));
8	f = pow(2.718282, e);
9	gau = 1 / sqrt(a);
10	gauss[gauske][i] = f * gau;
11	}

12	gauske++;
13	}

Fungsi perhitungan Gaussian pada tabel 5.14 di atas merupakan fungsi perhitungan berdasarkan pada **Persamaan (2.3)** sebelumnya. Fungsi ini mencari nilai Gaussian pada masing-masing fitur yang diuji yang kemudian akan diteruskan ke fungsi ProbabilityPosterior.

Pada fungsi ProbabilityPosterior memiliki beberapa parameter yang dibutuhkan yaitu nilai prior dari masing-masing kelas dan nilai Gaussian dari masing-masing fitur yang didapatkan. Implementasi kode program dari fungsi ProbabilityPosterior ini dapat dilihat pada Tabel 5.15.

Tabel 5.15 Kode Program Fungsi *ProbabilityPosterior*

Baris	Kode Program
1	void ProbabilityPosterior(float prior, int i) {
2	for(int j =0;j < 9;j++) {
3	if(j == 0) {
4	hasil[i] = (gauss[i][j] * 1000); //agar angka di belakang
5	koma tidak hilang
6	} else {
7	hasil[i] = hasil[i] * (gauss[i][j] * 1000);
8	}
9	}
10	hasil[i] = hasil[i] * prior;
	}

Pada Tabel 5.15 dapat dilihat untuk mencari nilai dari *Posterior* pada masing-masing kelas adalah dengan melakukan perhitungan perkalian antara nilai *Gaussian* pada masing-masing fitur yang didapatkan sebelumnya dengan nilai *prior* setiap kelas.

Setelah nilai *Posterior* masing-masing kelas berhasil didapatkan, tahap berikutnya yang dilakukan adalah menentukan kesimpulan. Implementasi kode program dari penentuan kesimpulan pada sistem ini dapat dilihat pada Tabel 5.16.

Tabel 5.16 Kode Program Penentuan Kesimpulan

Baris	Kode Program
1	void kesimpulan() {
2	for(int i = 0;i < 3;i++) {
3	if(i == 0) {
4	tertinggi = hasil[i];
5	index = i+1;
6	} else if(tertinggi < hasil[i]) {

7	tertinggi = hasil[i];
8	index = i + 1;
9	}
10	}
11	if (index == 1) {
12	sendDataToSerial('X', 1);
13	} else if (index == 2) {
14	sendDataToSerial('Y', 2);
15	} else if (index == 3) {
16	sendDataToSerial('Z', 3);
17	}
18	}

Pada Tabel 5.16 di atas dapat dilihat penarikan kesimpulan pada sistem ini adalah dengan melakukan perbandingan nilai *Posterior* pada masing-masing kelas dimana kelas dengan nilai *Posterior* terbesar merupakan hasil kesimpulan yang didapatkan. Pada baris ke-11 hingga baris ke-16, ketika hasil kesimpulan tersebut sudah didapatkan maka hasil kesimpulan tersebut akan dikirimkan ke Processing.

5.2.3.4 Implementasi Kode Program Pengiriman Data Arduino ke Processing

Begitu nilai dari R interval, BPM dan hasil kesimpulan telah didapatkan, Arduino akan mengirimkan nilai-nilai tersebut ke Processing. Fungsi dari aplikasi Processing ini adalah untuk menampilkan seluruh keluaran pada sistem. Implementasi dari kode program pengiriman data dari Arduino ke Processing dapat dilihat pada Tabel 5.17.

Tabel 5.17 Kode Program Pengiriman Data dari Arduino ke Processing

Baris	Kode Program
1	static int outputType = PROCESSING_VISUALIZER; // Untuk
2	menentukan output berada di processing, bukan di serial plotter
3	
4	void serialOutput() {
5	switch(outputType) {
6	case PROCESSING_VISUALIZER:
7	sendDataToSerial('S', Signal);
8	break;
9	case SERIAL_PLOTTER:
10	Serial.print(BPM);
11	Serial.print(",");
12	Serial.print(IBE);
13	Serial.print(",");
14	Serial.println(Signal);
15	break;
16	default:

```

17         break;
18     }
19 }
20 void serialOutputWhenBeatHappens() {
21     switch(outputType) {
22         case PROCESSING_VISUALIZER:
23             sendDataToSerial('B',BPM);    // mengirim BPM dengan prefix
24             'B'
25             sendDataToSerial('Q',IBI);    // mengirim IBI dengan prefix
26             'Q'
27             break;
28         default:
29             break;
30     }
31 }
32 ISR(TIMER2_COMPA_vect){
33     ...
34     sendDataToSerial('R',RR[indexRR]);    // mengirim nilai R
35     interval dengan prefix 'R'
36     ...
37 }
38 void kesimpulan() {
39     ...
40     if (index == 1) {
41         sendDataToSerial('X', 1);    // send time between beats with
42         a 'Q' prefix
43     } else if (index == 2) {
44         sendDataToSerial('Y', 2);    // send time between beats with
45         a 'Q' prefix
46     } else if (index == 3) {
47         sendDataToSerial('Z', 3);    // send time between beats with
48         a 'Q' prefix
49     }
50 }

```

Baris ke-1 pada Tabel 5.17 di atas merupakan inisialisasi sekaligus deklarasi pada variabel `outputType` dengan nilai `PROCESSING_VISUALIZER`. Dengan inisialisasi tersebut maka keluaran sistem akan berada di Processing, bukan di *serial monitor* Arduino. Setelah itu pada baris ke-4 hingga ke-17 merupakan kondisi ketika sistem menggunakan keluaran di Processing, maka sistem akan mengirimkan variabel `Signal` dengan prefix 'S' ke Processing yang berada di dalam fungsi `serialOutput()`. Kemudian pada baris ke-20 hingga baris ke-29, sistem akan mengirimkan nilai BPM dan IBI dengan prefix masing-masing 'B' dan 'Q' di dalam

fungsi `serialOutputWhenBeatHappens()`. Pada baris ke-32 hingga baris ke-35 sistem akan mengirimkan nilai terakhir yaitu nilai RR dengan prefix 'R' di dalam fungsi ISR. Pada baris ke-40 hingga baris ke-48 merupakan pengiriman hasil kesimpulan dengan menggunakan klasifikasi metode *Naïve Bayes*.

5.2.3.5 Implementasi Kode Program Menampilkan Hasil Keluaran Menggunakan Processing

Tahap ini dimaksudkan untuk melakukan implementasi kode program Processing yang digunakan untuk memberikan hasil keluaran Sistem Pendeteksi Premature Ventricular Contraction Aritmia. Tahap pertama yang dilakukan untuk melakukan implementasi kode program menampilkan hasil keluaran pada sistem ini adalah melakukan inisialisasi variabel yang dapat dilihat pada Tabel 5.18.

Tabel 5.18 Kode Program Inisialisasi Variabel Pada Processing

Baris	Kode Program
1	<code>import processing.serial.*;</code>
2	<code>PFont font;</code>
3	<code>PFont portsFont;</code>
4	<code>Scrollbar scaleBar;</code>
5	
6	<code>Serial port;</code>
7	
8	<code>int Sensor; // menyimpan data sensor ad8232 dari arduino</code>
9	<code>int IBI; // menyimpan waktu IBI dari arduino</code>
10	<code>int BPM; // menyimpan nilai BPM dari arduino</code>
11	<code>int[] RR = new int[8];</code>
12	<code>int index = 0;</code>
13	<code>int[] RawY; // menyimpan data gelombang sebelum disesuaikan</code>
14	<code>ukurannya</code>
15	<code>int[] ScaledY; // digunakan untuk menentukan ukuran gelombang</code>
16	<code>int[] rate; // digunakan untuk menentukan posisi bpm</code>
17	
18	<code>String decision = "";</code>
19	<code>color eggshell = color(255, 253, 248);</code>
20	<code>int heart = 0; // menyimpan nilai untuk menentukan jantung</code>
21	<code>berdetak atau tidak</code>
22	
23	<code>int PulseWindowWidth = 490;</code>
24	<code>int PulseWindowHeight = 512;</code>
25	<code>int BPMWindowWidth = 180;</code>
26	<code>int BPMWindowHeight = 340;</code>
27	<code>boolean beat = false</code>
28	
29	<code>String serialPort;</code>

30	<code>String[] serialPorts = new String[Serial.list().length];</code>
31	<code>boolean serialPortFound = false;</code>
32	<code>Radio[] button = new Radio[Serial.list().length];</code>

Baris ke-1 pada Tabel 5.18 di atas merupakan command yang digunakan untuk mengambil data dari Arduino ke Processing. Kemudian baris ke-2 hingga baris ke-4 merupakan inisialisasi untuk melakukan pengaturan font yang digunakan dan scrollbar. Setelah itu baris ke-6 digunakan untuk mencari port serial yang digunakan oleh Arduino Uno. Baris ke-8 hingga baris ke-21 merupakan inisialisasi yang memiliki fungsi untuk menyimpan nilai-nilai data yang dikirimkan dari Arduino dan pengaturan ukuran untuk ditampilkan. Pada baris ke-23 hingga baris 27 merupakan pengaturan ukuran panjang dan lebar untuk menampilkan grafik dan BPM. Terakhir pada baris ke-29 hingga baris ke-32 digunakan untuk menampilkan serial yang dapat digunakan dan ditampilkan menggunakan radio button.

Setelah inisialisasi selesai dilakukan, sistem perlu melakukan pengaturan untuk mengatur tampilan dan ukuran dari sistem. Implementasi kode program untuk pengaturan ukuran tampilan sistem dapat dilihat pada Tabel 5.19.

Tabel 5.19 Kode Program Pengaturan Tampilan Awal Pada Processing

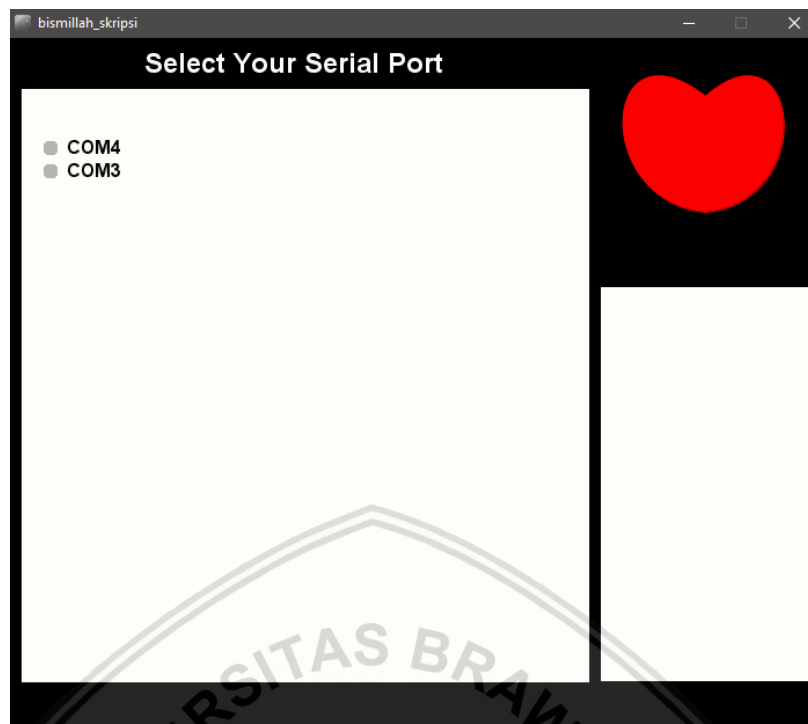
Baris	Kode Program
1	<code>void setup() {</code>
2	<code> size(700, 600); // Stage size</code>
3	<code> frameRate(100);</code>
4	<code> font = loadFont("Arial-BoldMT-24.vlw");</code>
5	<code> textFont(font);</code>
6	<code> textAlign(CENTER);</code>
7	<code> rectMode(CENTER);</code>
8	<code> ellipseMode(CENTER);</code>
9	
10	<code> RawY = new int[PulseWindowWidth];</code>
11	<code> ScaledY = new int[PulseWindowWidth];</code>
12	<code> rate = new int [BPMWindowWidth];</code>
13	<code> zoom = 0.75;</code>
14	
15	<code> fill(eggshell);</code>
16	<code> text("Select Your Serial Port",245,30);</code>
17	<code> listAvailablePorts();</code>
18	<code> ...</code>
19	<code>}</code>
20	<code>void listAvailablePorts(){</code>
21	


```

22     println(Serial.list());    // print a list of available serial
23     ports to the console
24     serialPorts = Serial.list();
25     fill(0);
26     textFont(font,16);
27     textAlign(LEFT);
28     // set a counter to list the ports backwards
29     int yPos = 0;
30     for(int i=serialPorts.length-1; i>=0; i--){
31         button[i] = new Radio(35,
32         95+(yPos*20),12,color(180),color(80),color(255),i,button);
33         text(serialPorts[i],50, 100+(yPos*20));
34         yPos++;
35     }
36     textFont(font);
37     textAlign(CENTER);
    }

```

Pada fungsi setup pada Tabel 5.19 di atas, baris ke-2 hingga baris ke-8 merupakan pengaturan untuk menentukan ukuran dari interface sistem, pengaturan pada font dan pengaturan paragraf yang akan ditampilkan. Kemudian pada baris ke-10 hingga baris ke-13 merupakan pengaturan untuk ukuran jendela grafik dan BPM yang akan ditampilkan. Baris ke-15 hingga baris ke-36 merupakan fungsi yang digunakan untuk menampilkan port serial yang dipakai oleh Arduino Uno. Tampilan awal pada Sistem Pendeteksi Premature Ventricular Contraction Aritmia ini ditunjukkan pada Gambar 5.15.



Gambar 5.15 Tampilan Awal Sistem Menggunakan Processing

Begitu Processing menemukan *port* yang digunakan pada Arduino Uno, Processing dapat menerima data-data yang telah dikirimkan oleh Arduino. Implementasi kode program yang digunakan untuk menerima data dari Arduino dapat dilihat pada Tabel 5.20.

Tabel 5.20 Kode Program Pengambilan Data Dari Arduino Ke Processing

Baris	Kode Program
1	<code>void serialEvent(Serial port){</code>
2	<code>try{</code>
3	<code>String inData = port.readStringUntil('\n');</code>
4	<code>inData = trim(inData);</code>
5	
6	<code>if (inData.charAt(0) == 'S'){</code>
7	<code>inData = inData.substring(1);</code>
8	<code>Sensor = int(inData);</code>
9	<code>}</code>
10	<code>if (inData.charAt(0) == 'B'){</code>
11	<code>inData = inData.substring(1);</code>
12	<code>BPM = int(inData);</code>
13	<code>beat = true;</code>
14	<code>heart = 20;</code>
15	<code>}</code>
16	<code>if (inData.charAt(0) == 'Q'){</code>
17	<code>inData = inData.substring(1);</code>

```

18     IBI = int(inData);
19     }
20     if (inData.charAt(0) == 'R'){
21         for (int i=0; i<9; i++){
22             inData = inData.substring(1);
23             RR[i] = int(inData);
24             if(i == 8) {
25                 i = 0;
26             }
27         }
28     }
29     if (inData.charAt(0) == 'X'){
30         decision = "Normal";
31     }
32     if (inData.charAt(0) == 'Y'){
33         decision = "Bigeminy";
34     }
35     if (inData.charAt(0) == 'Z'){
36         decision = "Trigeminy";
37     }
38 }catch(Exception e) {
39     }
40 }

```

Pada Tabel 5.20 dapat dilihat kode program pengambilan data dari Arduino ke Processing berada pada fungsi *serialEvent* dengan parameter *Serial port*. Pada baris ke-3 dan ke-4 merupakan kode yang digunakan untuk mengambil seluruh data dari Arduino ke Processing kemudian dilakukan *filtering* untuk memotong apabila terdapat spasi pada data yang dikirimkan. Pada baris ke-6 hingga baris ke-9 merupakan kondisi untuk mengambil data yang dikirimkan dari Arduino apabila data pertama yang dikirimkan memiliki label "S", maka data tersebut merupakan data sensor. Pada baris ke-10 hingga baris ke-15 merupakan kondisi untuk mengambil data yang dikirimkan dari Arduino apabila data pertama yang dikirimkan memiliki label "B", maka data tersebut merupakan data BPM. Pada baris ke-16 hingga baris ke-19 merupakan kondisi untuk mengambil data yang dikirimkan dari Arduino apabila data pertama yang dikirimkan memiliki label "Q", maka data tersebut merupakan data IBI. Pada baris ke-20 hingga baris ke-28 merupakan kondisi untuk mengambil data yang dikirimkan dari Arduino apabila data pertama yang dikirimkan memiliki label "R", maka data tersebut merupakan data R interval mulai dari RR_{t-7} hingga RR_t . Pada baris ke-29 hingga baris ke-37 merupakan kondisi untuk mengambil data hasil kesimpulan dari Arduino. Apabila data pertama yang dikirimkan memiliki label "X", maka data tersebut memiliki hasil kesimpulan "Normal", apabila data pertama yang dikirimkan memiliki label "Y", maka data tersebut memiliki hasil kesimpulan "Bigeminy" dan apabila data

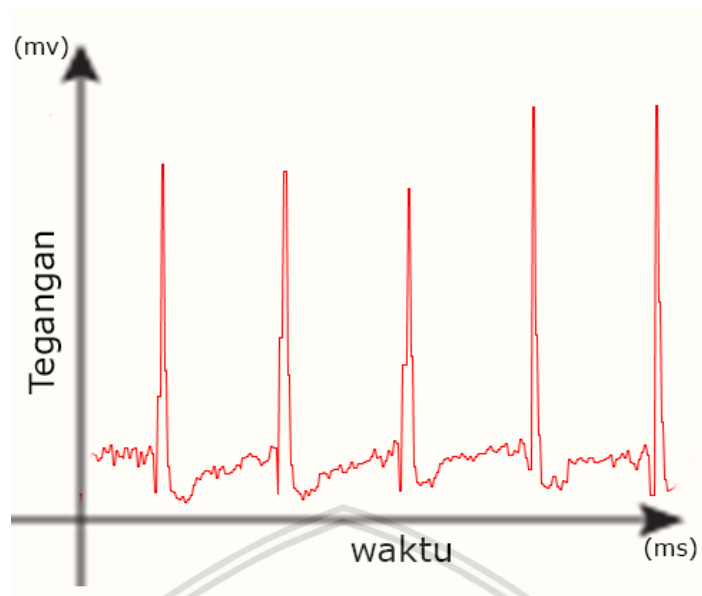
pertama yang dikirimkan memiliki label “Z”, maka data tersebut memiliki hasil kesimpulan “Trigeminy”.

Setelah seluruh data dapat diterima, Processing dapat menampilkan data-data tersebut berupa grafik dan angka. Data yang pertama kali ditampilkan oleh Processing adalah tampilan grafik EKG yang diambil dari data sensor. Implementasi kode program yang digunakan untuk menampilkan grafik EKG pada Processing dapat dilihat pada Tabel 5.21 berikut.

Tabel 5.21 Kode Program Menampilkan Grafik EKG Pada Processing

Baris	Kode Program
1	<code>void drawPulseWaveform() {</code>
2	<code>RawY[RawY.length-1] = (1023 - Sensor) - 212;</code>
3	<code>offset = map(zoom,0.5,1,150,0);</code>
4	<code>for (int i = 0; i < RawY.length-1; i++) {</code>
5	<code>RawY[i] = RawY[i+1];</code>
6	<code>float dummy = RawY[i] * zoom + offset;</code>
7	<code>ScaledY[i] = constrain(int(dummy),44,556);</code>
8	<code>}</code>
9	<code>stroke(250,0,0);</code>
10	<code>noFill();</code>
11	<code>beginShape();</code>
12	<code>for (int x = 1; x < ScaledY.length-1; x++) {</code>
13	<code>vertex(x+10, ScaledY[x]);</code>
14	<code>}</code>
15	<code>endShape();</code>
16	<code>}</code>

Pada Tabel 5.21 dapat dilihat kode program untuk menampilkan grafik EKG pada Processing berada pada fungsi `drawPulseWaveForm`. Baris ke-2 dan baris ke-3 berfungsi untuk menempatkan data “mentah” di akhir array `RawY`. Kemudian baris ke-4 hingga baris ke-10 digunakan untuk melakukan pengaturan ukuran dan warna grafik, `stroke(255,0,0)` yang berarti warna pada grafik adalah merah. Baris ke-11 hingga baris ke-15 digunakan untuk menggambarkan grafik dari data sensor yang didapat. Tampilan dari grafik EKG pada Processing dapat dilihat pada Gambar 5.16.



Gambar 5.16 Tampilan Grafik EKG Pada Processing

Untuk menampilkan nilai BPM dan R interval pada Sistem Pendeteksi Premature Ventricular Contraction Aritmia ini, sistem harus terlebih dahulu mendapatkan nilai dari BPM dan R interval yang telah diproses di Arduino sebelumnya. Implementasi kode program yang digunakan untuk menampilkan nilai fitur yang berupa R interval dan BPM pada Processing dapat dilihat pada Tabel 5.22.

Tabel 5.22 Kode Program Menampilkan Nilai Fitur Pada Processing

Baris	Kode Program
1	<code>void draw() {</code>
2	<code>if(serialPortFound){</code>
3	<code>...</code>
4	<code>drawBPMwaveform();</code>
5	<code>drawHeart();</code>
6	
7	<code>fill(eggshell);</code>
8	<code>text("Arrhythmia Detection",245,30);</code>
9	<code>text("IBI " + IBI + "mS",620,585);</code>
10	<code>text(BPM + " BPM",580,200);</code>
11	
12	<code>textSize(12);</code>
13	<code>textAlign(LEFT);</code>
14	<code>text("RRt-7 : " + RR[0] + " " + "RRt-6 : " + RR[1] + " " +</code>
15	<code>"RRt-5 : " + RR[2] + " " + "RRt-4 : " + RR[3] + " " + "RRt-3 :</code>
16	<code>" + RR[4] + " " + "RRt-2 : " + RR[5] + " " + "RRt-1 : " +</code>
17	<code>RR[6] + " " + "RRt : " + RR[7] + " ", 10, 585);</code>
18	

```

19 void drawBPMwaveform(){
20     if (beat == true){
21         beat = false;
22         for (int i=0; i<rate.length-1; i++){
23             rate[i] = rate[i+1];
24         }
25
26         BPM = min(BPM,200);
27         float dummy = map(BPM,0,200,555,215);
28         rate[rate.length-1] = int(dummy);
29     }
30     stroke(250,0,0);
31     strokeWeight(2);
32     noFill();
33     beginShape();
34     for (int i=0; i < rate.length-1; i++){
35         vertex(i+510, rate[i]);
36     }
37     endShape();
38 }
39
40 void drawHeart(){
41     fill(250,0,0);
42     stroke(250,0,0);
43     heart--;
44     heart = max(heart,0);
45     if (heart > 0){
46         strokeWeight(8);
47     }
48     smooth();
49     bezier(width-100,50, width-20,-20, width,140, width-
50 100,150);
51     bezier(width-100,50, width-190,-20, width-200,140, width-
52 100,150);
53     strokeWeight(1);
54 }

```

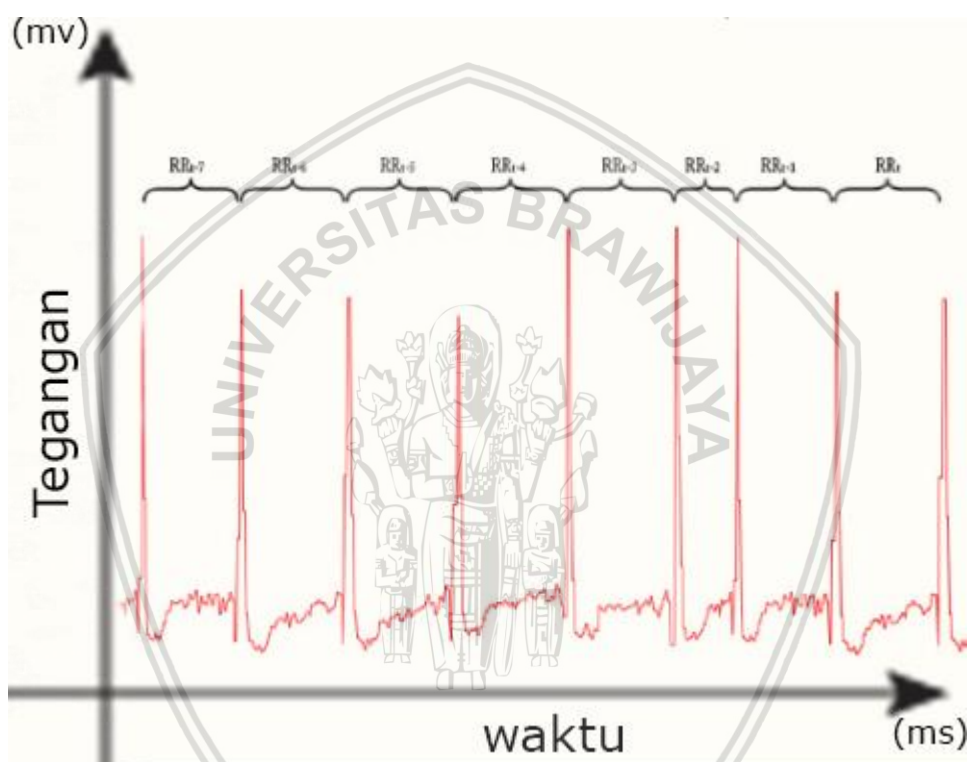
Pada Tabel 5.22, dapat dilihat bahwa terdapat tiga fungsi yang digunakan untuk menampilkan fitur klasifikasi *Naïve Bayes* yang digunakan pada sistem ini yaitu fungsi draw(), fungsi drawBPMwaveform() dan fungsi drawHeart(). Fungsi draw() digunakan untuk menampilkan tulisan dan nilai dari R interval, BPM dan IBI. Fungsi drawBPMwaveform() merupakan fungsi yang digunakan untuk menggambarkan grafik BPM sehingga mudah terlihat perbedaan nilai antara nilai

BPM saat ini dengan nilai BPM sebelumnya. Fungsi drawHeart() merupakan fungsi yang digunakan untuk menggambarkan gambar hati diatas tulisan BPM, dan membuat gambar hati tersebut berdetak ketika terdapat detak jantung yang terdeteksi. Tampilan dari fitur R interval klasifikasi *Naïve Bayes* yang digunakan pada sistem ini dapat dilihat pada Gambar 5.17.

RRt-7 : 786 RRt-6 : 818 RRt-5 : 844 RRt-4 : 864 RRt-3 : 866 RRt-2 : 872 RRt-1 : 850 RRt : 798

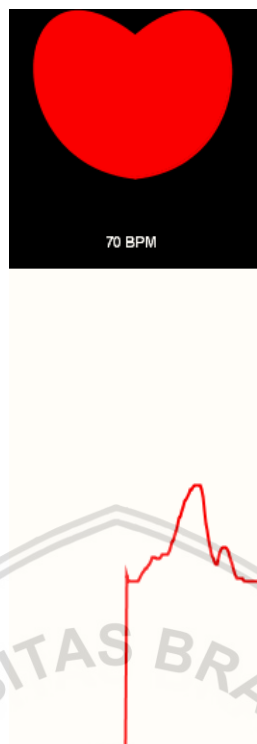
Gambar 5.17 Tampilan Fitur R Interval Pada Processing

Dengan keterangan grafik dari masing-masing R interval tersebut dapat dilihat pada Gambar 5.18.



Gambar 5.18 Fitur R Interval pada Grafik EKG

Pada Gambar 5.18 terdapat RRt-7 yang merupakan grafik waktu interval detak jantung yang pertama, kemudian RRt-6 yang merupakan grafik waktu interval detak jantung yang kedua dan seterusnya. Kemudian tampilan dari fitur BPM klasifikasi *Naïve Bayes* yang digunakan pada sistem ini dapat dilihat pada Gambar 5.19.



Gambar 5.19 Tampilan Fitur BPM Pada Processing

Terdapat tiga tampilan untuk menampilkan fitur BPM pada Processing ini, yaitu tampilan gambar hati untuk mengetahui jantung berdetak, tampilan nilai dari BPM dan tampilan grafik dari nilai BPM. Setelah nilai dari setiap fitur didapatkan, maka sistem dapat mengambil hasil kesimpulan dari Arduino dan ditampilkan melalui Processing. Implementasi kode program yang digunakan untuk menampilkan hasil kesimpulan pada Processing dapat dilihat pada Tabel 5.23.

Tabel 5.23 Kode Program Menampilkan Hasil Kesimpulan Klasifikasi Metode *Naïve Bayes* Pada Processing

Baris	Kode Program
1	<code>void draw() {</code>
2	<code>...</code>
3	<code>textSize(12);</code>
4	<code>textAlign(LEFT);</code>
5	<code>text("Decision : " + decision,460,585);</code>
6	<code>...</code>
7	<code>}</code>

Pada Tabel 5.23 di atas dapat dilihat untuk menampilkan hasil kesimpulan klasifikasi metode *Naïve Bayes* pada Processing berada dalam fungsi *draw()* dimana nilai dari variabel *decision* bergantung pada hasil perhitungan yang didapatkan dari Arduino. Hasil kesimpulan ini akan berubah setiap delapan kali nilai R interval dan BPM didapatkan, karena Sistem Pendeteksi *Premature*

Ventricular Contraction Aritmia ini berjalan secara *real time*. Tampilan dari hasil kesimpulan klasifikasi metode *Naïve Bayes* pada Processing ini dapat dilihat pada Gambar 5.20.

Decision : Normal

Gambar 5.20 Tampilan Hasil Kesimpulan Klasifikasi *Naïve Bayes* Pada Processing



BAB 6 PENGUJIAN DAN ANALISIS

Pada pengujian dan analisis ini membahas mengenai proses pengujian dan melakukan analisis dari pengujian yang dilakukan sebelumnya berdasarkan sistem yang dibuat. Tujuan dari pengujian ini adalah untuk mengetahui apabila sistem telah memenuhi seluruh kebutuhan yang ingin dicapai. Terdapat tiga pengujian yang dilakukan pada sistem ini, yaitu pengujian akurasi perhitungan BPM pada sistem yang dibandingkan dengan manual, pengujian akurasi dari *Naïve Bayes* dan pengujian waktu komputasi pemrosesan yang dilakukan oleh sistem.

6.1 Pengujian Akurasi Perhitungan BPM

Pengujian akurasi perhitungan BPM ini dilakukan dengan cara membandingkan hasil deteksi manual dengan hasil deteksi pada sistem. Deteksi manual dilakukan dengan meletakkan jari pada bagian leher sampai detak jantung dapat terdeteksi, kemudian dihitung detak jantung tersebut selama satu menit. Deteksi sistem dilakukan oleh pembacaan sensor AD8232 yang diproses oleh Arduino Uno.

6.1.1 Tujuan Pengujian

Tujuan dari pengujian akurasi perhitungan BPM atau jumlah detak jantung selama satu menit pada Sistem Pendeteksi Premature Ventricular Contraction Aritmia ini adalah untuk mengetahui ketepatan hasil deteksi BPM yang dilakukan oleh sistem dibandingkan dengan deteksi manual.

6.1.2 Prosedur Pengujian

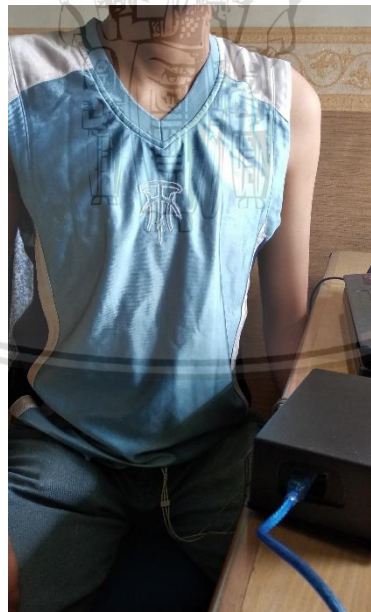
Prosedur yang dilakukan untuk melakukan pengujian akurasi perhitungan BPM pada Sistem Pendeteksi Premature Ventricular Contraction Aritmia antara lain sebagai berikut :

1. Menghubungkan Arduino Uno dengan laptop.
2. Memasang elektroda sesuai dengan perancangan sebelumnya kepada objek yang berbeda-beda.
3. Menjalankan kode program Arduino yang digunakan pada sistem.
4. Menjalankan kode program Processing yang digunakan pada sistem.
5. Mencatat hasil nilai BPM yang didapatkan oleh deteksi sistem.
6. Mengukur nilai BPM yang didapatkan dengan menggunakan deteksi manual, yaitu menempatkan jari pada leher bagian atas sampai terdeteksi detak jantung kemudian dihitung selama 1 menit seperti pada Gambar 6.1.



Gambar 6.1 Pendeteksian BPM Menggunakan Deteksi Manual

7. Mencatat hasil perhitungan BPM menggunakan deteksi manual, kemudian membandingkan nilai tersebut dengan nilai BPM yang didapatkan menggunakan deteksi oleh sistem. Pendeteksian nilai BPM yang dilakukan oleh sistem ini dapat dilihat pada Gambar 6.2 dimana pada dalam baju tersebut telah terpasang elektroda dengan berdasarkan perancangan sebelumnya.



Gambar 6.2 Pendeteksian BPM Menggunakan Deteksi dari Sistem

8. Menghitung nilai presentase *error* dari nilai BPM yang didapatkan oleh sistem dengan nilai BPM yang didapatkan dengan deteksi manual menggunakan **Persamaan (6.1)**.

$$error = \frac{|pengukuran\ BPM\ manual - pengukuran\ BPM\ oleh\ sistem|}{pengukuran\ BPM\ manual} \times 100\% \quad (6.1)$$

6.1.3 Hasil dan Analisis Pengujian

Tabel 6.1 Hasil Pengujian Perbandingan Nilai Pendeteksian BPM Manual dengan Pendeteksian BPM oleh Sistem

Pengujian ke-	Nilai BPM		Error (%)
	Manual	Sistem	
1	70	72	2,857
2	70	68	2,857
3	79	72	8,860
4	78	72	7,692
5	63	60	4,762
6	76	71	6,579
7	65	59	9,231
8	79	72	8,860
9	64	61	4,688
10	68	61	10,294
11	72	67	6,944
12	84	107	27,380
13	71	66	7,042
14	68	60	11,764
Rata-rata			8,558

Dari Tabel 6.1, terdapat nilai *error* yang didapatkan menggunakan perhitungan **Persamaan (6.1)** pada setiap pengujian. Contoh perhitungan dari persentase *error* pada pengujian ke-1 adalah sebagai berikut :

$$error = \frac{|pengukuran\ BPM\ manual - pengukuran\ BPM\ oleh\ sistem|}{pengukuran\ BPM\ manual} \times 100\%$$

$$error = \frac{|70-72|}{70} \times 100\%$$

$$error = 2,857\%$$

Kemudian diperoleh rata-rata *error* dari seluruh pengujian yaitu sebesar 8,558%. Untuk menghitung nilai rata-rata dari nilai *error* seluruh pengujian adalah sebagai berikut :

$$\begin{aligned}
 \text{Rata - rata error} &= \frac{\text{total persentase error}}{\text{jumlah pengujian}} \\
 &= \frac{2,857 + 2,857 + 8,86 + 7,692 + 4,762 + 6,579 + 9,231 + 8,86 + 4,688 + 10,294 + 6,944 + 27,38 + 7,042 + 11,764}{14} \\
 &= \frac{119,41}{14} \\
 &= 8,558\%
 \end{aligned}$$

Error yang didapatkan dari sistem ini biasanya terjadi karena adanya perubahan ritme dari detak jantung pada responden. Kemudian terdapat nilai error yang besar pada pengujian ke-12 yang terjadi karena responden memiliki tingkat obesitas yang tinggi.

6.2 Pengujian Akurasi Hasil Klasifikasi *Naïve Bayes*

Sistem pendeteksi *Premature Ventricular Contraction* Aritmia ini diperlukan untuk dapat melakukan klasifikasi jenis PVC Aritmia dengan berdasarkan fitur dari sistem yaitu nilai dari R Interval dan nilai BPM. Kemudian sistem akan membandingkan nilai fitur tersebut dengan nilai fitur yang ada pada data latih.

6.2.1 Tujuan Pengujian

Tujuan dilakukan pengujian akurasi hasil klasifikasi *Naïve Bayes* adalah untuk mengetahui tingkat akurasi dari penggunaan metode *Naïve Bayes* pada Sistem Pendeteksi *Premature Ventricular Contraction* Aritmia ini.

6.2.2 Prosedur Pengujian

Prosedur yang akan dilakukan untuk melakukan pengujian akurasi *Naïve Bayes* pada sistem ini antara lain sebagai berikut :

1. Menghubungkan Arduino Uno dengan laptop menggunakan kabel serial.
2. Memasang elektroda sesuai dengan perancangan sebelumnya kepada objek yang berbeda-beda.
3. Menjalankan kode program Arduino yang digunakan pada sistem.
4. Menjalankan kode program Processing yang digunakan pada sistem.
5. Menunggu sistem berhasil membaca nilai R interval sebanyak delapan kali.
6. Ketika sistem sudah membaca nilai R interval sebanyak delapan kali, hasil klasifikasi *Naïve Bayes* pada sistem akan ditampilkan di bagian bawah pada tampilan antarmuka Processing.
7. Menghitung nilai akurasi dari hasil pengujian yang didapatkan menggunakan **Persamaan (6.2)**.

$$\text{Akurasi} = \frac{\text{Total data} - \text{Jumlah data yang tidak sesuai}}{\text{Total data}} \times 100\% \quad (6.2)$$

6.2.3 Hasil dan Analisis Pengujian

Tabel 6.2 Data dan Hasil Pengujian Klasifikasi *Naïve Bayes*

No	RR _{t-7}	RR _{t-6}	RR _{t-5}	RR _{t-4}	RR _{t-3}	RR _{t-2}	RR _{t-1}	RR _t	BP M	Kelas	Hasil Sistem	Keses uaian
1	1000	906	940	974	1002	894	902	1006	61	Normal	Normal	Sesuai
2	814	811	789	792	789	817	653	994	71	Normal	Normal	Sesuai
3	786	806	756	761	786	806	842	853	72	Normal	Normal	Sesuai
4	252	265	362	475	263	475	362	252	107	Normal	<i>Bigeminy</i>	Tidak Sesuai
5	881	833	833	842	833	911	908	917	70	Normal	Normal	Sesuai
7	983	981	1022	1006	1019	1031	997	1033	59	Normal	Normal	Sesuai
8	556	1075	547	1217	606	1231	633	1100	61	<i>Bigeminy</i>	<i>Bigeminy</i>	Sesuai
9	950	1022	1152	902	926	896	1002	1042	60	Normal	Normal	Sesuai
10	512	1264	806	535	864	796	498	894	78	<i>Trigemin y</i>	<i>Trigemin y</i>	Sesuai
11	736	720	794	874	716	730	792	814	72	Normal	Normal	Sesuai
12	778	786	828	806	760	814	844	550	67	Normal	Normal	Sesuai
13	806	868	934	854	820	862	892	832	60	Normal	Normal	Sesuai
14	580	1062	512	1006	480	1016	576	970	72	<i>Bigeminy</i>	<i>Bigeminy</i>	Sesuai

Contoh perhitungan dari hasil klasifikasi *Naïve Bayes* dari sistem yang telah memiliki nilai prior, mean dan standar deviasi pada perancangan sebelumnya dengan menggunakan data uji pengujian ke-1 pada Tabel 6.2 di atas adalah sebagai berikut.

1. Melakukan perhitungan nilai *Gaussian* dari masing-masing fitur yang telah diujikan (RR_{t-7}=1000, RR_{t-6}=906, RR_{t-5}=940, RR_{t-4}=974, RR_{t-3}=1002, RR_{t-2}=894, RR_{t-1}=902, RR_t=1006, BPM=61)

$$P(RR_{t-7} = 1000 \mid \text{Normal}) = \frac{1}{\sqrt{2\pi\sigma_{RR_{t-7}(\text{Normal})}^2}} e^{-\frac{(1000 - \bar{x}_{RR_{t-7}(\text{Normal})})^2}{2\sigma_{RR_{t-7}(\text{Normal})}^2}}$$

$$= \frac{1}{\sqrt{2 \times 3,14 \times (94,28629)^2}} e^{-\frac{(1000 - 794,416667)^2}{2 \times (94,28629)^2}}$$

$$= 0,00036834$$

$$P(RR_{t-7} = 1000 \mid \text{bigeminy}) = \frac{1}{\sqrt{2\pi\sigma_{RR_{t-7}(\text{bigeminy})}^2}} e^{-\frac{(1000 - \bar{x}_{RR_{t-7}(\text{bigeminy})})^2}{2\sigma_{RR_{t-7}(\text{bigeminy})}^2}}$$

$$= \frac{1}{\sqrt{2 \times 3,14 \times (79,15736)^2}} e^{-\frac{(1000 - 522,535714)^2}{2 \times (79,15736)^2}}$$

$$= 6,3394 \times 10^{-11}$$

$$P(RR_{t-7} = 1000 \mid \text{trigeminy}) = \frac{1}{\sqrt{2\pi\sigma_{RR_{t-7}(\text{trigeminy})}^2}} e^{-\frac{(1000 - \bar{x}_{RR_{t-7}(\text{trigeminy})})^2}{2\sigma_{RR_{t-7}(\text{trigeminy})}^2}}$$

$$= \frac{1}{\sqrt{2 \times 3,14 \times (99,64078)^2}} e^{-\frac{(1000 - 562,5714286)^2}{2 \times (99,64078)^2}}$$

$$= 2,61754 \times 10^{-11}$$

Perhitungan yang sama dilakukan untuk mencari nilai peluang *Gaussian* pada setiap fitur RR_{t-6} , RR_{t-5} , RR_{t-4} , RR_{t-3} , RR_{t-2} , RR_{t-1} , RR_t dan BPM. Nilai dari hasil seluruh perhitungan *Gaussian* dapat ditunjukkan pada Tabel 6.3.

Tabel 6.3 Nilai *Gaussian* Pada Setiap Jenis Fitur dan Kelas Aritmia

	Normal	Bigeminy	Trigeminy
RR_{t-7}	0,000334956	$6,3394 \times 10^{-11}$	$2,61574 \times 10^{-7}$
RR_{t-6}	0,002398292	0,002948713	$5,33248 \times 10^{-5}$
RR_{t-5}	0,001469597	$3,73974 \times 10^{-12}$	0,003468379
RR_{t-4}	0,001111404	0,003137703	$1,77575 \times 10^{-16}$
RR_{t-3}	0,000884803	$1,2758 \times 10^{-14}$	0,001052844
RR_{t-2}	0,002530478	0,002709902	0,003781933
RR_{t-1}	0,001974709	$6,56588 \times 10^{-10}$	0,000616766
RR_t	0,000814091	0,001898927	0,001432376
BPM	0,013733266	0,000126986	0,035940536

2. Menghitung nilai *Posterior* pada masing-masing jenis kelas *Premature Ventricular Contraction* Aritmia

$$\begin{aligned}
 &P(\text{Normal} | RR_{t-7}=1000, RR_{t-6}=906, RR_{t-5}=940, RR_{t-4}=974, RR_{t-3}=1002, RR_{t-2}=894, RR_{t-1}=902, \\
 &RR_{t=1006}, BPM_{=61}) \\
 &= P_{\text{Normal}} \times P(RR_{t-7} = 1000 | \text{Normal}) \times P(RR_{t-6} = 906 | \text{Normal}) \times P(RR_{t-5} = 940 | \text{Normal}) \times \\
 &P(RR_{t-4} = 974 | \text{Normal}) \times P(RR_{t-3} = 1002 | \text{Normal}) \times P(RR_{t-2} = 894 | \text{Normal}) \times P(RR_{t-1} \\
 &= 902 | \text{Normal}) \times P(RR_t = 1006 | \text{Normal}) \times P(BPM = 61 | \text{Normal}) \\
 &= 0,255319149 \times 0,00036834 \times 0,002538776 \times 0,001605636 \times 0,001280459 \times 0,001059007 \\
 &\times 0,00269149 \times 0,002101929 \times 0,000823216 \times 0,014907631 \\
 &= 3,60911.10^{-26} \\
 \\
 &P(\text{bigeminy} | RR_{t-7}=1000, RR_{t-6}=906, RR_{t-5}=940, RR_{t-4}=974, RR_{t-3}=1002, RR_{t-2}=894, RR_{t-1}=902, \\
 &RR_{t=1006}, BPM_{=61}) \\
 &= P_{\text{bigeminy}} \times P(RR_{t-7} = 1000 | \text{bigeminy}) \times P(RR_{t-6} = 906 | \text{bigeminy}) \times P(RR_{t-5} = 940 | \\
 &\text{bigeminy}) \times P(RR_{t-4} = 974 | \text{bigeminy}) \times P(RR_{t-3} = 1002 | \text{bigeminy}) \times P(RR_{t-2} = 894 \\
 &| \text{bigeminy}) \times P(RR_{t-1} = 902 | \text{bigeminy}) \times P(RR_t = 1006 | \text{bigeminy}) \times \\
 &P(BPM = 61 | \text{bigeminy}) \\
 &= 6,3394.10^{-11} \times 0,002948713 \times 3,73974.10^{-12} \times 0,003137703 \times 1,2758.10^{-14} \times \\
 &0,002709902 \times 6,56588.10^{-10} \times 0,001898927 \times 0,000126986 \\
 &= 7,15296.10^{-60} \\
 \\
 &P(\text{trigeminy} | RR_{t-7}=1000, RR_{t-6}=906, RR_{t-5}=940, RR_{t-4}=974, RR_{t-3}=1002, RR_{t-2}=894, RR_{t-1}=902, \\
 &RR_{t=1006}, BPM_{=61}) \\
 &= P_{\text{trigeminy}} \times P(RR_{t-7} = 1000 | \text{trigeminy}) \times P(RR_{t-6} = 906 | \text{trigeminy}) \times P(RR_{t-5} = 940 | \\
 &\text{trigeminy}) \times P(RR_{t-4} = 974 | \text{trigeminy}) \times P(RR_{t-3} = 1002 | \text{trigeminy}) \times P(RR_{t-2} = 894 \\
 &| \text{trigeminy}) \times P(RR_{t-1} = 902 | \text{trigeminy}) \times P(RR_t = 1006 | \text{trigeminy}) \times \\
 &P(BPM = 61 | \text{trigeminy}) \\
 &= 2,61574.10^{-7} \times 5,33248.10^{-5} \times 0,003468379 \times 1,7757510^{-16} \times 0,001052844 \times \\
 &0,003781933 \times 0,000616766 \times 0,001432376 \times 0,035940536 \\
 &= 1,61761.10^{-43}
 \end{aligned}$$

Pada hasil perhitungan nilai peluang *Posterior* di atas, jenis *Premature Ventricular Contraction* Aritmia yang mempunyai nilai peluang *Posterior* tertinggi adalah normal dengan nilai *Posterior* sebesar $3,60911.10^{-26}$.

Dari 14 data yang telah diuji terdapat 1 hasil sistem yang tidak sesuai dengan kelas aslinya. Dari pengujian tersebut, didapatkan nilai akurasi dari sistem berdasarkan **Persamaan (6.2)** adalah 92,857% dengan proses perhitungan sebagai berikut :

$$Akurasi = \frac{\text{Total data} - \text{Jumlah data yang tidak sesuai}}{\text{Total data}} \times 100\%$$

$$Akurasi = \frac{14-1}{14} \times 100\%$$

$$Akurasi = 92,857\%$$

6.3 Pengujian Waktu Komputasi Pemrosesan Sistem

Pengujian waktu komputasi pemrosesan sistem ini dilakukan dengan cara menghitung waktu yang diperlukan oleh sistem mulai dari sistem mengambil data pada sensor hingga sistem mendapatkan hasil klasifikasi dengan menggunakan fungsi `millis()` yang ada pada Arduino.

6.3.1 Tujuan Pengujian

Tujuan dilaksanakannya pengujian waktu komputasi pemrosesan pada sistem ini adalah untuk mengetahui waktu yang dibutuhkan oleh sistem dalam melakukan proses perhitungan klasifikasi Premature Ventricular Contraction Aritmia dengan menggunakan *Naïve Bayes*. Hal ini digunakan untuk mengetahui waktu yang diperlukan oleh sistem untuk mendapatkan hasil klasifikasi.

6.3.2 Prosedur Pengujian

Prosedur yang dilakukan untuk melakukan pengujian waktu komputasi pada sistem antara lain sebagai berikut :

1. Menghubungkan Arduino Uno dengan laptop.
2. Memasang elektroda sesuai dengan perancangan sebelumnya kepada objek yang berbeda-beda.
3. Menjalankan kode program Arduino yang digunakan untuk menghitung waktu komputasi sistem menggunakan kode program pada Tabel 6.4.

Tabel 6.4 Implementasi Kode Program Arduino Untuk Menghitung Waktu Komputasi Sistem

Baris	Kode Program
1	<code>volatile unsigned long waktu1, waktu2, waktuakhir;</code>
2	<code>void loop() {</code>
3	<code> waktu1=millis();</code>
4	<code> Serial.print("komputasi awal : ");</code>
5	<code> Serial.print(waktu1);</code>
6	<code> Serial.println(" ms");</code>
7	<code> ...</code>
8	<code> //fungsi pengambilang klasifikasi <i>Naïve Bayes</i></code>
9	<code> ...</code>
10	<code> waktu2 = millis();</code>
11	<code> waktuakhir = waktu2-waktu1;</code>
12	<code> Serial.print("waktu komputasi : ");</code>
13	<code> Serial.print(waktuakhir);</code>
14	<code> Serial.println(" ms");</code>

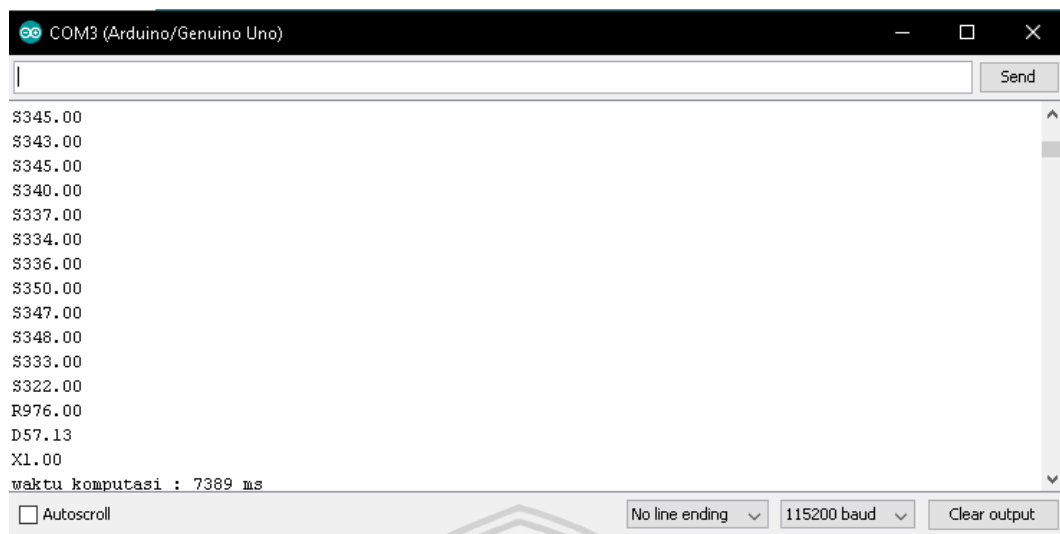
4. Membuka *serial monitor* yang terdapat pada Arduino IDE, kemudian waktu komputasi akan ditampilkan di *serial monitor* tersebut.

6.3.3 Hasil dan Analisis Pengujian

Tabel 6.5 Hasil Pengujian Waktu Komputasi Sistem

Pengujian ke-	Waktu Komputasi (ms)
1	7839
2	7288
3	6470
4	7045
5	6969
6	6839
7	6701
8	7353
9	7836
10	7731
11	7030
12	7515
13	6890
14	6726
Rata-rata	7159,429

Waktu komputasi pada sistem merupakan waktu komputasi mulai dari sistem membaca nilai pada sensor hingga sistem mendapatkan hasil klasifikasi. Berdasarkan hasil pengujian waktu komputasi yang dilakukan sebanyak 14 kali pada Tabel 6.4 di atas, sistem ini memiliki nilai rata-rata waktu komputasi sebesar sekitar 7,2 detik. Nilai tersebut didapatkan karena untuk mendapatkan hasil klasifikasi, sistem memerlukan jantung responden berdetak sebanyak 8 kali terlebih dahulu untuk mendapatkan kesimpulan hasil klasifikasi. Tampilan dari salah satu hasil pengujian waktu komputasi sistem ditunjukkan pada Gambar 6.3.



Gambar 6.3 Hasil Pengujian Waktu Komputasi Sistem



BAB 7 PENUTUP

Bab ini berisi mengenai penarikan kesimpulan berdasarkan tahap-tahap yang telah dilakukan sebelumnya dan penyampaian saran dari penulis yang diharapkan dapat digunakan untuk pengembangan penelitian yang serupa dengan penelitian ini selanjutnya.

7.1 Kesimpulan

Berdasarkan dari rumusan masalah yang diajukan pada awal penelitian dan hasil analisis dari pengujian yang telah dilakukan maka dapat diambil kesimpulan sebagai berikut :

1. Sensor ECG Module AD8232 Arduino dapat membaca nilai aktivitas listrik pada jantung manusia dengan nilai rata-rata *error* sebesar 8,558% untuk melakukan perhitungan BPM. Selain itu, sensor juga dapat membaca waktu interval setiap kali jantung berdetak dengan satuan ms, sehingga dapat dikatakan sensor berfungsi dengan baik karena dapat membaca perubahan nilai pada jantung setiap 2 ms atau 0,002 detik. Namun, sensor tidak dapat membaca aktivitas listrik jantung pada seseorang yang memiliki tingkat obesitas yang tinggi.
2. Pada penelitian ini telah dibuat sistem otomasi untuk melakukan deteksi jenis *Premature Ventricular Contraction* Aritmia berdasarkan waktu interval ketika jantung berdetak (R interval) dan jumlah detak jantung yang berdetak selama satu menit (BPM) menggunakan metode *Naïve Bayes* dengan hasil klasifikasi *Premature Ventricular Contraction* Aritmia menjadi tiga jenis, yaitu normal, *bigeminy* dan *trigeminy*. Akurasi yang didapatkan dari sistem yang telah diuji menggunakan 46 data latih dan 14 data uji ini adalah sebesar 92,857%.
3. Waktu rata-rata yang diperlukan oleh sistem ini mulai dari membaca sensor hingga mendapatkan hasil klasifikasi adalah sekitar 7,2 detik dari 14 kali pengujian.

7.2 Saran

Terdapat saran yang dapat dijadikan acuan untuk pengembangan penelitian ini ataupun penelitian selanjutnya adalah sebagai berikut :

1. Menggunakan metode klasifikasi lain untuk melakukan perbandingan dan menarik kesimpulan metode mana yang memiliki tingkat akurasi lebih tinggi.
2. Melakukan teknik pemasangan elektroda lain seperti 12-Lead atau lain sebagainya sehingga sensor dapat membaca termasuk kepada orang yang memiliki tingkat obesitas tinggi.
3. Menerapkan sistem serupa dengan tampilan antarmuka menggunakan aplikasi *mobile* sehingga pengguna dapat lebih nyaman dalam menggunakan sistem.

DAFTAR PUSTAKA

- Arduino. (2016). Retrieved from <http://www.arduino.org/products/boards/arduino-uno>
- Charles A. Herzag, J. M. (2008). Sudden Cardiac Death and Dialysis Patients. *Non-Coronary Heart Disease in Dialysis Patients*, 300-308.
- Clinical Education, UNM Hospitals. (2017, January). *Basic Arrhythmia*. Retrieved from Basic Arrhythmia: https://learningcentral.health.unm.edu/learning/user/onlineaccess/CE/bac_online/idio/index.html#nogo2
- Departemen Kardiologi & Kedokteran Vaskular FKUI. (2004). Buku Ajar Kardiologi. In D. K. FKUI, *Buku Ajar Kardiologi* (p. 65). Jakarta: Sagung Seto.
- Eduardo José da S.Luz, W. R.-C. (2016). Computer Methods and Programs in Biomedicine. *ECG-Based heartbeat classification for arrhythmia detection*, 144-164.
- EKG Indonesia. (2015, May). *Premature Ventricular Contraction / Ventrikular Ekstrasistol*. Retrieved from EKG Indonesia: <http://www.ina-ecg.com/2015/05/premature-ventricular-contraction.html>
- Encyclopedia Britannica. (2008, January 11). *Electrocardiography*. Retrieved from Encyclopedia Britannica: <https://www.britannica.com/science/electrocardiography>
- Fajar, R. (2017, November 6). *4 Jenis Pemeriksaan untuk Diagnosis Aritmia*. Retrieved from 1Health: <http://www.1health.id/id/article/category/sehat-a-z/4-jenis-pemeriksaan-untuk-diagnosis-aritmia.html>
- Fredy, F. C. (2017, June 2). *Aritmia*. Retrieved from Kerjanya: <http://www.kerjanya.net/faq/4352-aritmia.html>
- Goldberger AL, A. L.-K. (2000). PhysioBank, PhysioToolkit, and PhysioNet. *Components of a New Research Resource for Complex Physiologic Signals*.
- Heart Risk Warner. (2015). *Measure ECG, arrhythmia detection*. Retrieved from Heart Risk Warner: <http://www.hrwproject.com/ecg.html>
- K. Nanthakumar, V. S. (2017). Prediction of Ventricular Arrhythmia Using Naive Bayes Classifier. *International Journal of Advanced Research in Computer and Communication Engineering*, 185-196.
- Mayoclinic. (2018, February 8). *Premature ventricular contractions (PVCs)*. Retrieved from Mayoclinic: <https://www.mayoclinic.org/diseases-conditions/premature-ventricular-contractions/symptoms-causes/syc-20376757>
- Processing. (2014). *Welcome To Processing*. Retrieved from Processing: processing.org
- Wahyu Kusuma R, S. E. (2017). Rancang Bangun Alat Telemonitor Aritmia Jantung Berbasis Web. *SEMNASVOKTEK*, 161-167.
- World Health Organization. (2017, January). *The top 10 causes of death*. Retrieved from World Health Organization: <http://www.who.int/mediacentre/factsheets/fs310/en/>